



BELL & HOWELL SCHOOLS

DeVRY INSTITUTE OF TECHNOLOGY





BELL & HOWELL SCHOOLS

DeVRY INSTITUTE OF TECHNOLOGY

LOGIC CIRCUITS

COURSE 52



IMPORTANT NOTICE

Course 52 contains one large table. It is located directly below this sheet. It is Table 3 of Lesson 5230, titled "Coding and Decoding Circuits."

After you have finished studying the material in this volume, you may wish to file this table with the lesson to which it refers. To make this easy, the table has the lesson name in the upper right-hand corner and the page number in the lower left-hand corner.

REPORT BY NOTICE

1. The purpose of this report is to provide a summary of the information received from the various sources mentioned in the notice.

2. The information received from the various sources is as follows:

LOGIC CIRCUITS

CONTENTS

CONCEPTS OF SWITCHING LOGIC	Lesson 5205
SWITCHING CHARACTERISTICS	Lesson 5210
DIODE LOGIC CIRCUITS	Lesson 5215
ACTIVE SWITCHING CIRCUITS	Lesson 5220
BASIC COMBINATIONAL LOGIC	Lesson 5225
CODING AND DECODING CIRCUITS	Lesson 5230
BASIC ARITHMETIC CIRCUITS	Lesson 5235
DIGITAL INTEGRATED CIRCUITS	Lesson 5240
REGISTER AND COUNTER CIRCUITS	Lesson 5245
DATA STORAGE CIRCUITS AND DEVICES	Lesson 5250
UNIT EXAMINATION	COURSE 52

LOGIC CIRCUITS

CHAPTER 1

1.1 Introduction

1.2 Basic Logic Operations

1.3 Boolean Algebra

1.4 Logic Gates

1.5 Combinational Logic

1.6 Sequential Logic

1.7 Digital Systems

1.8 Summary

LOGIC CIRCUITS

CONTENTS

1. Introduction	1
2. Boolean Algebra	10
3. Logic Gates	25
4. Combinational Logic	45
5. Sequential Logic	65
6. Timing Diagrams	85
7. Memory	105
8. Arithmetic Logic Units	125
9. Microprocessors	145
10. Conclusion	165

DECIMAL	BINARY					
	F	E	D	C	B	A
	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
0	0	0	0	0	0	0
1	0	0	0	0	0	1
2	0	0	0	0	1	0
3	0	0	0	0	1	1
4	0	0	0	1	0	0
5	0	0	0	1	0	1
6	0	0	0	1	1	0
7	0	0	0	1	1	1
8	0	0	1	0	0	0
9	0	0	1	0	0	1
10	0	0	1	0	1	0
11	0	0	1	0	1	1
12	0	0	1	1	0	0
13	0	0	1	1	0	1
14	0	0	1	1	1	0
15	0	0	1	1	1	1
16	0	1	0	0	0	0

DECIMAL	BINARY					
	F	E	D	C	B	A
	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
17	0	1	0	0	0	1
18	0	1	0	0	1	0
19	0	1	0	0	1	1
20	0	1	0	1	0	0
21	0	1	0	1	0	1
22	0	1	0	1	1	0
23	0	1	0	1	1	1
24	0	1	1	0	0	0
25	0	1	1	0	0	1
26	0	1	1	0	1	0
27	0	1	1	0	1	1
28	0	1	1	1	0	0
29	0	1	1	1	0	1
30	0	1	1	1	1	0
31	0	1	1	1	1	1
32	1	0	0	0	0	0
33	1	0	0	0	0	1

DECIMAL	BINARY					
	F	E	D	C	B	A
	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
34	1	0	0	0	1	0
35	1	0	0	0	1	1
36	1	0	0	1	0	0
37	1	0	0	1	0	1
38	1	0	0	1	1	0
39	1	0	0	1	1	1
40	1	0	1	0	0	0
41	1	0	1	0	0	1
42	1	0	1	0	1	0
43	1	0	1	0	1	1
44	1	0	1	1	0	0
45	1	0	1	1	0	1
46	1	0	1	1	1	0
47	1	0	1	1	1	1
48	1	1	0	0	0	0
49	1	1	0	0	0	1
50	1	1	0	0	1	0

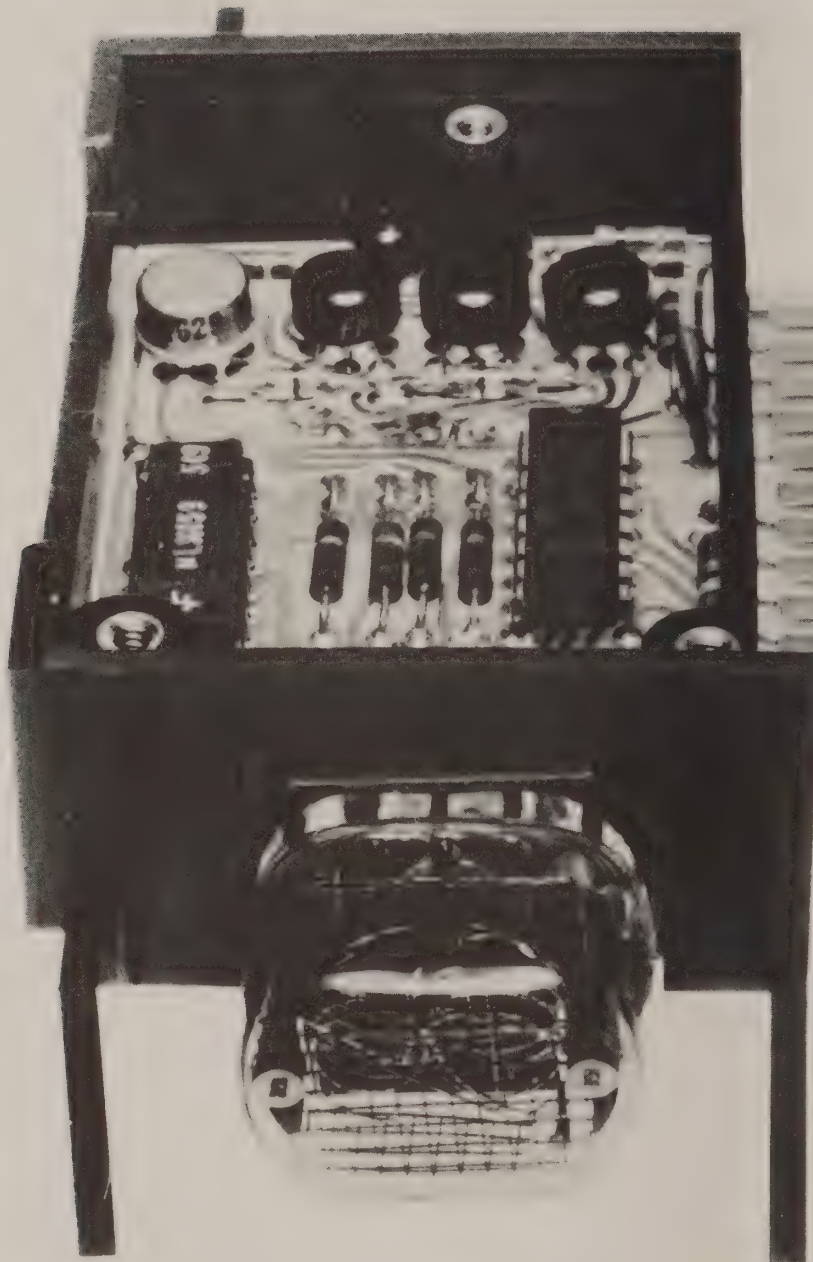
Table 3

CONCEPTS OF SWITCHING LOGIC

5205

© COPYRIGHT 1973
BELL & HOWELL SCHOOLS, INC.
ALL RIGHTS RESERVED





This low cost digital counter module is a complete package consisting of integrated circuit counting, the required logic, storage, binary-coded decimal, display driver circuits and a display tube.

Courtesy Fairchild Instruments

CONTENTS

Semiconductor Switching States	Page 4
Response Times	Page 7
Unsaturated States	Page 9
Active Region Switching	Page 10
Logical Gating Operations	Page 10
Basic Boolean Algebra	Page 11
OR and AND Functions	Page 14
The NOT Function	Page 17
NOR and NAND Functions	Page 18
The INHIBIT Function	Page 21
Positive and Negative Logic	Page 22
Multiple-Input Representations	Page 23
Appendix—MIL-STD-806B	Page 28

*A man never discloses his own
character so clearly as when he
discredits another's.*

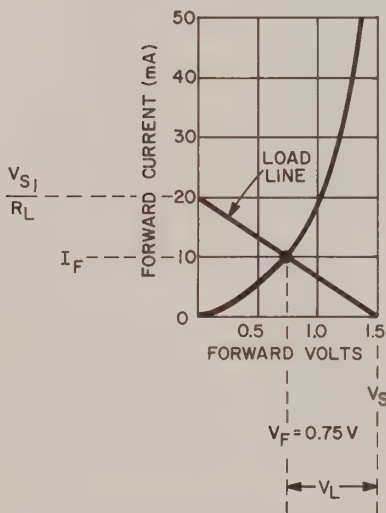
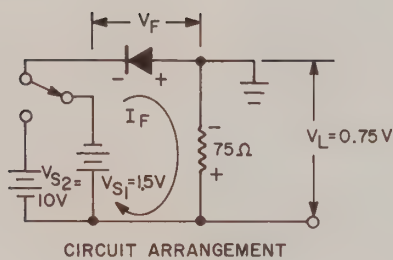
—Jean Paul Richter

CONCEPTS OF SWITCHING LOGIC

An electronic **GATING CIRCUIT** provides an output pulse, or the absence of an output pulse, depending on the voltages present at the various input terminals. In essence, the gate operates in an "on" state (conducting) or "off" state (nonconducting). Hence, the term gating circuit is synonymous with **SWITCHING CIRCUIT**.

Digital computers, modern telephone systems, complete industrial manufacturing and material processing facilities, and vast military command and control networks utilize complex switching systems in order to perform their intended functions economically, efficiently, and accurately. Each "switch" in these systems is a simple electronic circuit arrangement operating in an "on" or "off" state. Through the proper combination of these simple devices, it is possible to achieve almost any desired function. Establishing control operations on the basis of input conditions is known as **LOGICAL SWITCHING**. The logical switching on and off states can be related to logical decisions such as "yes" and "no", or "true" and "false".

In this lesson, we will introduce the basic concepts of **SWITCHING LOGIC**, known also as **DIGITAL LOGIC** or **BINARY LOGIC** because of the two-state (on and off) representation of circuit conditions.



LOAD LINE ANALYSIS

CONDITIONS FOR THE DIODE
"ON" STATE

Figure
1

SEMICONDUCTOR SWITCHING STATES

Modern switching systems may utilize hundreds, even thousands, of diodes, transistors or other solid state (semiconductor) devices. For most applications, these devices will be driven very rapidly from one extreme condition (cutoff) to the opposite extreme condition (saturation). Before describing how these two states are used to represent logical operations, we will describe the electrical conditions for these states.

In switching applications the active device, or electronic switch, is used in conjunction with a resistor or a resistor network. In either case, the equivalent circuit, the electronic switch, in series with a fixed resistor value, is that of a variable resistor.

The variable resistor exhibits one of two values: an extremely small value when the device is "on" or an extremely large value when the device is "off".

Figure 1 illustrates the operating conditions of a forward biased junction diode. The conducting region of the diode, for the operating point shown

where the forward diode current, I_F , is 10 mA and the forward diode voltage drop, V_F , is 0.75 volt, displays a resistance of 75Ω . A specific voltage drop and a forward current flow describe the "on" state of the diode. Figure 2 illustrates the "off" state of the diode. This is the reverse biased region. However, for switching applications, the reverse voltage cannot be allowed to exceed the Zener breakdown value. For the conditions shown, the diode resistance is on the order of $800\text{ k}\Omega$. There is a relatively high voltage drop

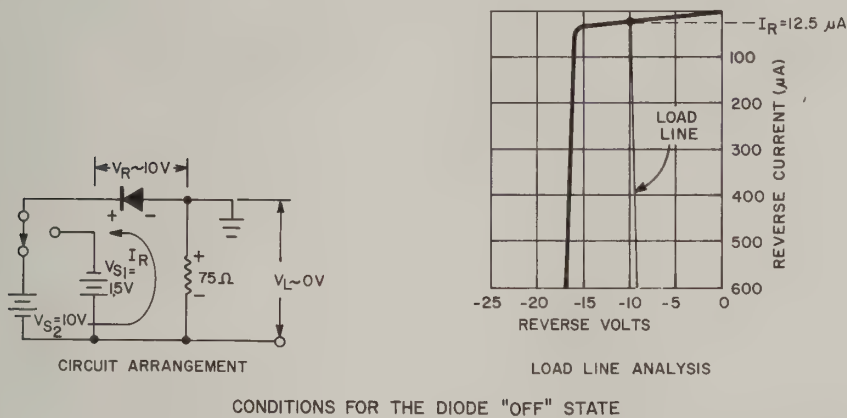


Figure 2

across the diode, but very little current flow. This set of conditions defines the "off" state of the diode. Although the exact conditions for each state vary from one diode to another, and from one circuit to another for a given diode, the magnitude of the "on" and "off" states generally will be similar to the values shown in Figures 1 and 2.

A characteristic curve typical of tunnel diodes is shown in Figure 3. The "on" state corresponds to the peak point on the curve; where the resistance of the device is 55 ohms, the voltage drop is low, and current is relatively high. The "off" state corresponds to the valley point on the curve; where the resistance of the device is on the order of $3\text{ k}\Omega$, the voltage drop is high, and current is low.

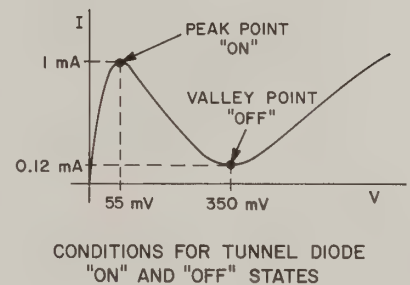


Figure 3

The four-layer diode of PNPN structure, as shown in Figure 4, has a characteristic somewhat like that of the tunnel diode. The breakover point (high voltage drop and low current flow) corresponds to the "off" condition. The resistance is on the order of $30\text{ k}\Omega$. The holding point (low voltage drop and high current flow), corresponds to the "on" condition. The resistance is on the order of 950 ohms .

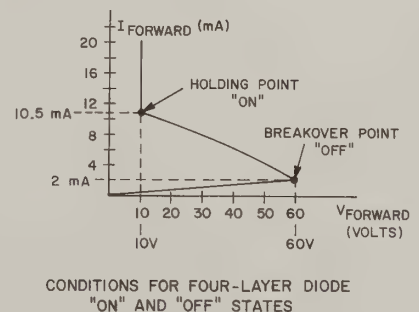


Figure 4

Transistors are also used in logical switching applications. A basic circuit for the junction transistor switch is shown in Figure 5, along with a typical

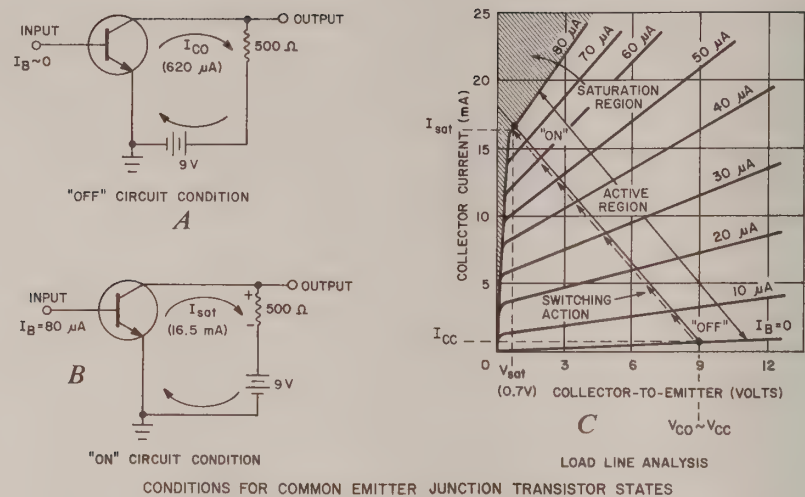


Figure 5

set of characteristics. For the "off" condition, the collector voltage is essentially the same as the supply voltage and the current is extremely small. The resistance is approximately 14 kΩ for the condition shown in Figure 5A. The "on" condition is associated with transistor saturation. The collector voltage is small; the current is high. Resistance is less than 45 ohms. Unlike the two-element diode devices, the transistor also has input conditions associated with the "on" and "off" states. Cutoff is obtained by reverse biasing the base-to-emitter junction; saturation, by heavily forward biasing this junction. The PNP transistor (not shown) has similar general characteristics, with input and output voltage polarities reversed.

The unijunction transistor (UJT) has characteristic "on" and "off" conditions functionally similar to those of the tunnel and four-layer diodes. These conditions are shown in Figure 6. However, the UJT is a three-element device, like the junction transistor. The "off" condition corresponds to the

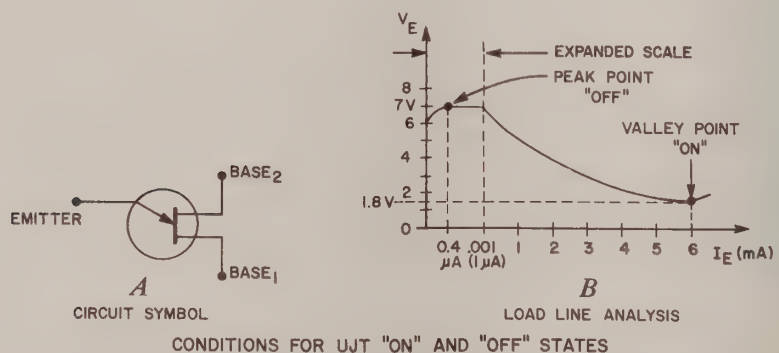
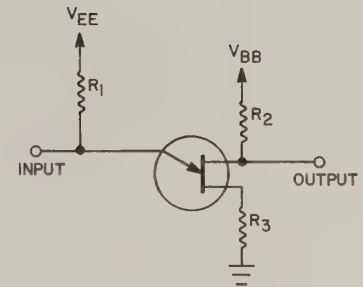


Figure 6

peak point: a high voltage drop, a very low current, and a resistance on the order of $17.5 \text{ M}\Omega$. The "on" condition corresponds to the valley point: a low voltage drop, high current, and a resistance of 300 ohms. Since it is a three element device, the input must be used to switch the output branch of the circuit, as shown in Figure 7.

The enhancement mode metal-oxide-semiconductor field effect transistor (MOSFET), shown in Figure 8, is most commonly used in FET switching circuits. This is because it can be switched with single-polarity pulses. The "off" condition represents a resistance on the order of $30 \text{ k}\Omega$; the "on" condition, less than 700 ohms. The device shown is an N-type channel type. P-channel devices are also available. P-channel devices have similar characteristics, except that all voltage polarities are reversed.



UJT SWITCHING CIRCUIT ARRANGEMENT

Figure 7

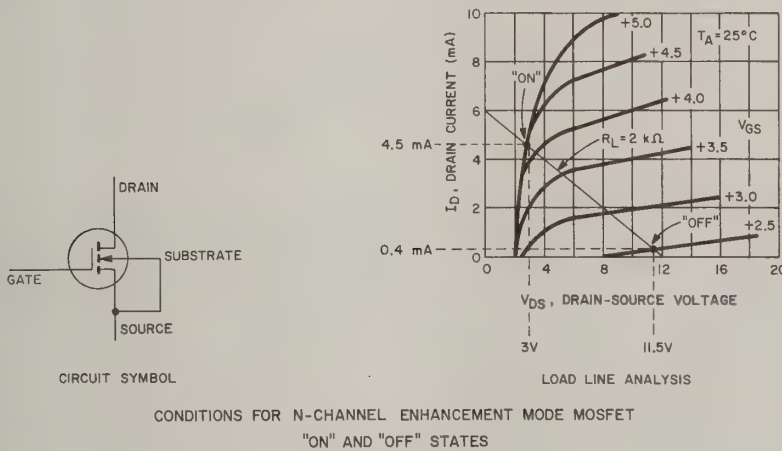


Figure 8

Response Times

In most cases logic circuit input and output pulse waveforms will be shown as rectangular pulses with very sharp leading and trailing edges. However, such "ideal" waveforms will not always be observed in actual practice. This is because all electronic devices (solid state or vacuum tube) produce inherent delays in their response to rapid voltage input variations. In addition, circuit inductances and capacitances may form integrating differentiating networks, producing further distortions in the waveforms.

In general, the pulse response of an electronic device is specified in terms of RISE TIME, FALL TIME and PROPAGATION DELAY TIME. Rise time and fall time are associated with the leading and trailing edges of a pulse, respectively. Both are measured between the 10% and 90% peak pulse amplitude points, as shown in Figure 9. Propagation delay is measured

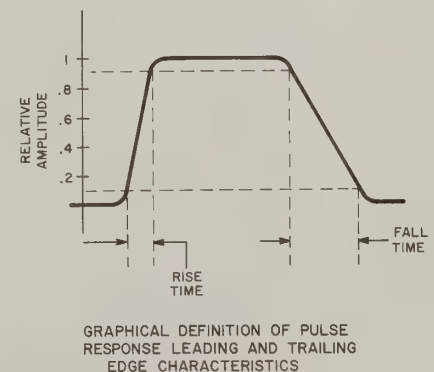


Figure 9

relative to the input and output pulses, using the 50% peak pulse amplitude points, as shown in Figure 10. Depending on the circuit arrangement, one of these delays is a TURN-ON DELAY for the device; the other, a TURN-OFF DELAY. The two are not necessarily equal.

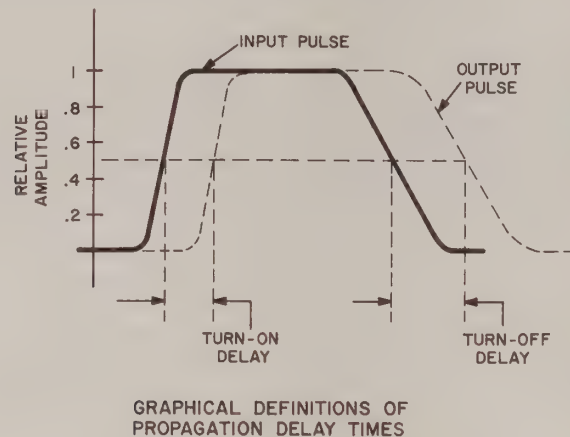


Figure 10

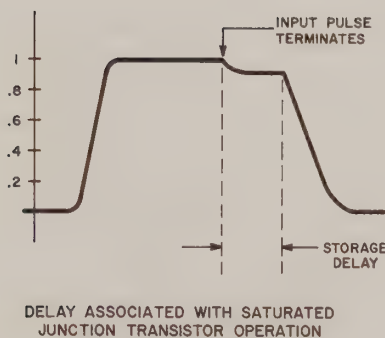


Figure
11

Another form of distortion, peculiar to junction transistors, is the STORAGE DELAY illustrated in Figure 11. This occurs when the device is brought out of saturation. When a junction transistor is in saturation, both the base-to-emitter and base-to-collector junctions are forward biased. Hence, both the emitter and collector regions act as "emitters", injecting majority carriers into the base region. Carrier density in the base region becomes extremely high. When the transistor comes out of saturation, the excess carriers must flow back into the collector region. In effect, the base region acts as a capacitor, storing a charge during saturation, then discharging as it comes out of saturation. The time required for the excess carriers to reach an equilibrium condition may be as great as one microsecond. The only way that this delay can be avoided in a junction transistor switching circuit is to prevent the device from going into saturation.

Networks external to an electronic device may also contribute to the total rise, fall, and propagation delay times associated with a circuit. In addition, the external networks may introduce distortions of the form shown in Figure 12. Actually, the networks producing these distortions may not always be apparent. Inductance and capacitance, for example, may be due to stray wiring or junction characteristics not shown in the circuit schematic. SAG in the flat, top portion of a pulse is associated with RC coupling networks. When sag is present, there is also an UNDERSHOOT, unless limiting circuitry is included to stop this phenomenon. However, undershoot may occur by itself when transformer coupling is used. OVERSHOOT may also be present with transformer coupling, or may be the result of an RLC branch within a network. (Remember, the lead inductance and/or stray and component capacitance is present along with the discrete components.) RING-

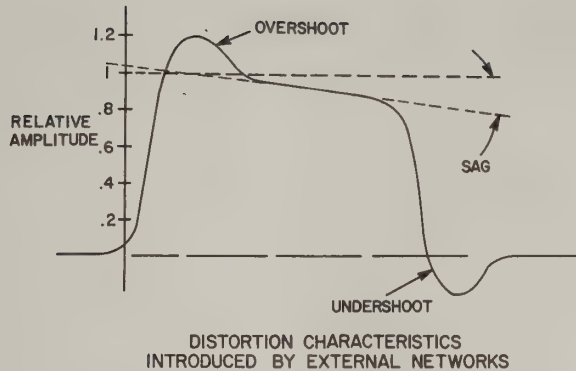


Figure 12

ING is an oscillation within an RLC path and may be an undershoot or overshoot phenomenon.

Unsaturated States

Eliminating the saturated condition of a transistorized switching circuit, allows transistors to be used at much higher switching rates. To accomplish this, the conducting level of the transistor is clamped, or limited, in such a way that the collector current stays below the saturation value. This places one state of operation in the active portion of the transistor characteristic, as shown in Figure 13.

In collector voltage clamping, as shown in Figure 14, V_D and D_1 are selected such that the collector-to-emitter voltage is prevented from dropping to a value equal to or less than the base-to-emitter voltage. When the transistor is cut off, the diode is reverse biased, acting as a large resistance in the output circuit. Hence, the diode has no effect on cutoff operation—the output voltage is near V_{CC} . When the transistor is driven out of cutoff, the transistor collector-to-emitter voltage begins to drop. In a saturated switch (When $V_{BE(sat)}$ is larger than $V_{CE(sat)}$, both junctions are forward biased, producing saturation), it would drop to almost zero. In this circuit, however, as soon as it drops to just below V_D , the diode is forward biased, and the collector-to-emitter voltage is clamped at this level. This method of clamping does not provide much improvement in response time. The collector current may reach a higher level than in a saturated switch using the same transistor.

Collector current clamping is shown in Figure 15. As long as the transistor is in cutoff, the diode is reverse biased. When the transistor comes out of cutoff and the collector-to-emitter voltage becomes less than the drop across the diode, the diode “conducts”, increasing the current through R_1 . There is an associated increase in voltage across R_1 , and a proportional decrease

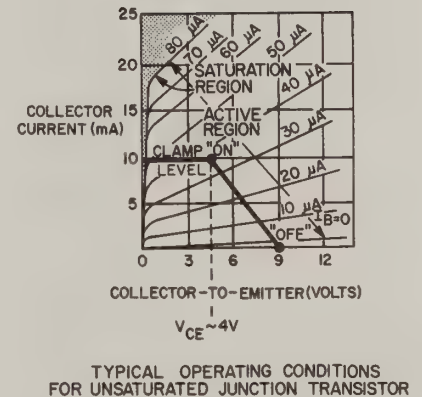


Figure 13

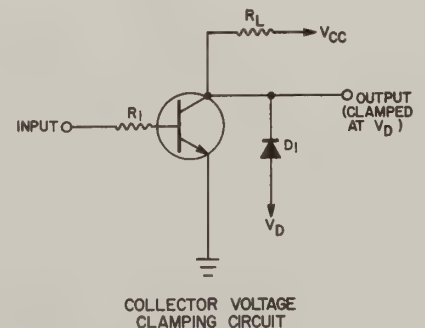


Figure 14

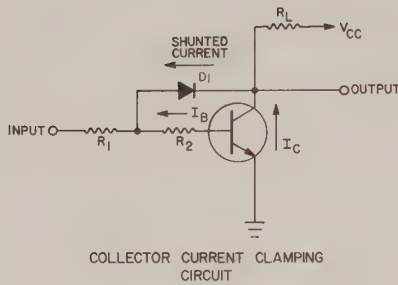


Figure 15

in voltage across R_2 . The base current to the transistor is controlled by the voltage across R_2 . Hence, I_B is "held down", as is I_C . For a desired value of collector current (less than the saturation value), R_2 is calculated as follows:

$$R_2 = \beta \left[\frac{V_{CC}}{I_C'} - R_L \right] \quad (1)$$

where I_C' is the clamped collector current.

PNP transistor switching can be controlled in exactly the manner described for NPN switching. However, all bias and input/output voltage polarities are reversed, as are the diode orientations within the circuit.

Active Region Switching

Unsaturated switches may only operate in the active region. The "off" condition is defined by some level of collector current flow and collector-to-emitter voltage in the active region as shown in Figure 16. However, achieving this combination of operating points requires a rather complex circuit design—often involving two transistors rather than only one transistor. When such a circuit is used, the output voltage variation no longer alternates between near zero and near V_{CC} . Instead, it takes on a "limited swing", as shown in Figure 17.

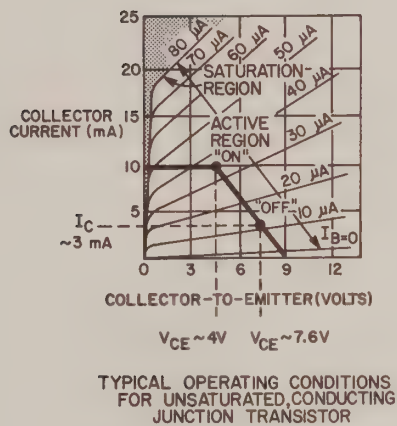


Figure 16

An advantage of having both the "on" and "off" conditions in the active region of the transistor is higher switching speed. Disadvantages (in addition to the more complex circuit design) are increased device power dissipation and reduced output voltage swing.

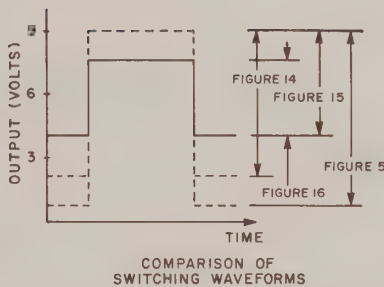


Figure 17

LOGICAL GATING OPERATIONS

You have undoubtedly made many decisions in your lifetime by debating the **LOGIC** of a situation and its various alternatives. Such logic, however, is frequently a matter of opinion and is somewhat hard to "pindown" in terms of an exact description. To structure complex electronic systems for logical operations, it is necessary that these operations be defined in formal and repeatable terms. In 1854, George Boole laid the foundation for developing such a formal method of representation. The rules are now known as **BOOLEAN ALGEBRA** or **LOGICAL ALGEBRA**. Without Boolean algebra (or its equivalent), the development of effective switching combinations and the full potential of digital system techniques would probably never be realized. There are three basic **LOGICAL FUNCTIONS**: **OR**,

The following Practice Exercise questions cover the subjects which you have just studied. They are:

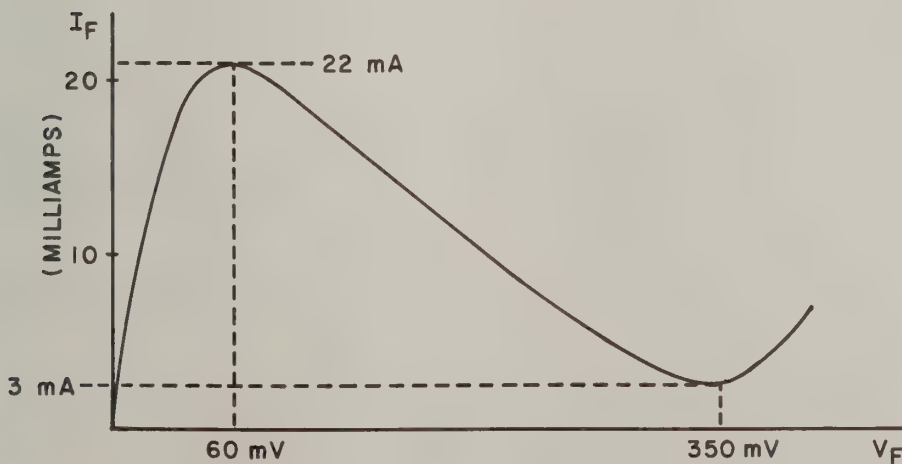
SEMICONDUCTOR SWITCHING STATES

RESPONSE TIMES

UNSATURATED STATES

ACTIVE REGION SWITCHING

1. What two states are common to all electronic switching devices?
2. The "off" state of an electronic switching device exhibits a _____ voltage drop across the device, a _____ current through the device, and a _____ resistance than the "on" state.
3. A junction diode is "on" when a _____ voltage is applied and is "off" when a _____ voltage is applied, unless this voltage exceeds the _____ value.
4. The figure below illustrates a typical tunnel diode characteristic. (a) What is the voltage drop across the device, and the current through the device, when it is "on"? (b) What is the "on" resistance of the device? (c) What is the voltage drop across the device, and current through the device, when it is "off"? (d) What is the "off" resistance of the device?



5. The voltage drop across the output terminals of a junction transistor or an FET in the “off” state in most circuits is nearly equal to the _____ voltage.
6. Why is the enhancement mode MOSFET used in most FET switching applications?

AND, and **NOT**. All other functions are achieved by the appropriate combination of these three.

LOGICAL GATING or logical switching, and the use of logical algebra, is based on the use of a two-state switching circuit, the identity between the “on” and “off” conditions, and logical decisions such as “yes” and “no” or “true” and “false”. The states may also be represented by the **LOGICAL DIGITS**, “one” and “zero”. Hence the complex capabilities of modern switching systems, including the digital computer, are actually based on the application of a very simple two-state concept, as shown in Table 1.

Although Table 1 presents the most common associations, there is no reason why these associations cannot be reversed. For example, no-false-zero may be associated with the “on” state; yes-true-one may be associated with the “off” state. In fact, in some situations it is more convenient to identify yes or true with zero and no or false with one. Any combination is permitted: as long as a consistent set of definitions is used throughout.

It must be emphasized that the logical 1 and logical 0 do not, themselves, refer to the numerical value of any quantity. **THEY ARE SIMPLY DIGITAL REPRESENTATIONS OF A SWITCHING STATE.** They are characterized by a signal voltage level (or, perhaps, by the absence of a signal voltage); even the selection of this voltage value is arbitrary. For example, a positive signal value of 30 volts might be selected to represent logical 1 in a particular system. Then, a less positive signal voltage level would be used to indicate logical 0. This might be 20 volts, 10 volts, –10 volts, –30 volts, or the absence of a voltage (any value less than 30 volts). Conversely, the logical 1 in another system might be represented by a –10 volt signal level. In this case, a level that is positive with respect to –10 volts would be used for the logical 0. This might be –5 volts, 15 volts, or the absence of a voltage, etc.

For many control purposes, the final output of a switching system may be a 1 or a 0—signaling another device (such as a motor) to turn “on” or turn “off”. However, most applications are more complex, requiring the use of a series of electronic switches to represent a numerical value. Such a conversion of actual numerical values into a sequence of logical digits is known as **DIGITAL ENCODING**.

Basic Boolean Algebra

In a particular industrial facility, it may be desirable to turn on a set of damper motors when either the temperature reaches 500°F or when a 10-minute time interval has elapsed. This is a simple set of logical conditions.

Physical State of an Electronic Device	On	Off
Associated Logical Representations	Yes	No
	True	False
	1	0

A Basis for Digital Operations

Table 1

In order to express these conditions in logical algebraic terms, it is necessary to represent each with a symbol:

Z = command to "turn on" damper motors
 A = temperature reaches 500°F
 B = 10-minute time interval elapsed.

The selection of symbols is completely arbitrary. With this set of definitions, however, the conditions described by the first sentence of this paragraph can be concisely presented as:

$$A + B = Z \quad (\text{OR}) \quad (2)$$

This is known as the OR function. The plus sign is a **LOGICAL CONNECTIVE** for the OR function and should not be confused with the arithmetic process of addition. The "equation" is read as "A or B equals Z". This implies that either of two signals, signal A indicating that the temperature has reached 500°F, OR signal B indicating that 10 minutes has elapsed, will generate a third signal (Z) to turn on the motors. The order of the symbols in the OR function has no significance. Hence:

$$A + B = B + A, \quad (3)$$

indicating that both expressions mean the same thing.

For the industrial application described, it also would be desirable when the temperature reaches 500°F at the same instant that the 10 minute time interval elapses, that the motors will still be turned on. With the same symbolic definitions, these conditions would be represented by the logical algebra statement:

$$A \cdot B = Z \quad (\text{AND}) \quad (4)$$

where the dot is the logical connective for the AND function and should not be confused with the arithmetic process of multiplication. The equation is read as "A and B equals Z". This implies that two simultaneous signals generate a third signal because the two necessary conditions have occurred. Frequently, the AND function is written without the connective dot:

$$AB = Z \quad (\text{AND}) \quad (4a)$$

This is still read in the same manner and, again, must not be confused with the arithmetic process of multiplication. The order of writing the two AND conditions is not important. Hence:

$$A \cdot B = B \cdot A \quad (5)$$

The complete set of conditions for operating the motors, described by both of the preceding paragraphs, is given by:

$$A + B + (A \cdot B) = Z \quad (\text{INCLUSIVE-OR}) \quad (6)$$

which is read, "A or B or A and B equals Z". This equation fully indicates that either of the two signals, or both the signals (simultaneously) will turn on the motors. This particular combination of OR and AND functions represents a situation frequently encountered in digital circuits and, in total, is classified as the **INCLUSIVE-OR** function. In fact, this combination occurs so frequently that the statement:

$$A + B = Z$$

has come to be "understood" as the **INCLUSIVE-OR** function without writing the associated AND term. However, if there is any chance of confusion in a given situation the complete expression for **INCLUSIVE-OR** should be used. You must also keep in mind the fact that $A + B$ can be interpreted as the **INCLUSIVE-OR** function (which "includes" the AND function as well as the OR function); under no circumstances can $A \cdot B$, by itself, be interpreted as anything more than the basic AND function.

It is convenient, especially in more complex situations, to allow for both the positive and the negative (true and false) of a particular condition. This permits the logic to include **NEGATION**, through the use of the NOT function as a condition of operation. This is also known as **INVERSION** or **COMPLEMENTATION**. Negation changes a logical quantity to its opposite state. Due to the lack of universal standards in Boolean algebra, there are many terms with the same meaning. Therefore, be sure to review the **IMPORTANT DEFINITIONS** section, where most of these are presented. The negation function will be represented by placing a bar over the letter symbol, and will always refer to the opposite state or condition compared to that of the symbol without the bar over it. Using the previous definitions:

$\bar{Z}$ = no command to "turn on" damper motors (that is, the motors remain "off")

$\bar{A}$ = temperature does not reach 500°F

$\bar{B}$ = 10-minute time interval does not elapse.

These symbols are read as "not Z", "not A", and "not B", respectively. The admission of the NOT function allows logical statements to be written for every possible combination of conditions which might be associated with the damper motors previously described. The original statement:

$$A + B = Z$$

indicates that either of the two conditions will turn on the motors. Hence, if condition A occurs and condition B does not occur, the motors still go "on":

$$A + \bar{B} = Z.$$

Input	Output
$A + B$	Z
$A + \bar{B}$	Z
$\bar{A} + B$	Z
$\bar{A} + \bar{B}$	$\bar{Z}$

Truth Table for
the Example
Logic Conditions

Table
2

OR	+	U	V
AND	·	∩	∧
NOT	-	'	~

Common Logical
Connective Symbols

Table
3

If condition B occurs and condition A does not occur, the motors still go "on":

$$\bar{A} + B = Z.$$

However, if neither of the conditions occurs, the motors do not go "on":

$$\bar{A} + \bar{B} = \bar{Z}.$$

This complete set of conditions and results for the given logical situation may be summarized to form a **TRUTH TABLE** as shown in Table 2.

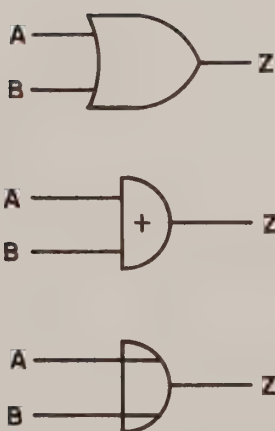
Table 3 presents a set of the various logical connectives that are frequently found in the literature. Others are also possible. Some manufacturers "invent" their own symbols. However, uncommon symbols are usually defined on the data sheets that those manufacturers provide. The symbols in the left-hand column are used throughout this course, although the dot may sometimes be omitted when the AND function is indicated.

OR and AND Functions

Figure 18 illustrates some of the more common logic symbols used to indicate the OR function on a **LOGIC DIAGRAM**. In Boolean algebra, this logic is indicated by:

$$A + B = Z$$

implying that either condition A or B (or A and B, if interpreted as INCLUSIVE-OR) will produce result Z. None of these representations show how an electronic circuit can be designed to actually accomplish the OR function. There are, of course, many alternatives. Figure 19 presents one possible circuit for achieving the OR function utilizing a diode arrangement. With both inputs maintained at a 0 volt level as in Figure 19A, the output will also be at a 0 volt level. If a positive step is applied to input A, as shown in Figure 19B, forward current will flow in the circuit, thus producing a positive output step. If a positive step is applied to input B, forward current will flow in the circuit, thus producing a positive output step. This is shown in Figure 19C. In fact, a positive step applied simultaneously at both input terminals will also cause a positive output step, as shown in Figure 19D; therefore, this is an INCLUSIVE-OR circuit.



SELECTION OF SYMBOLS
FOR THE OR FUNCTION

Figure
18

Table 2 summarized the input-output conditions for the OR functions using Boolean symbols. The digits 1 and 0 may also be associated with logical conditions (and with circuit input-output conditions). If the positive volt-

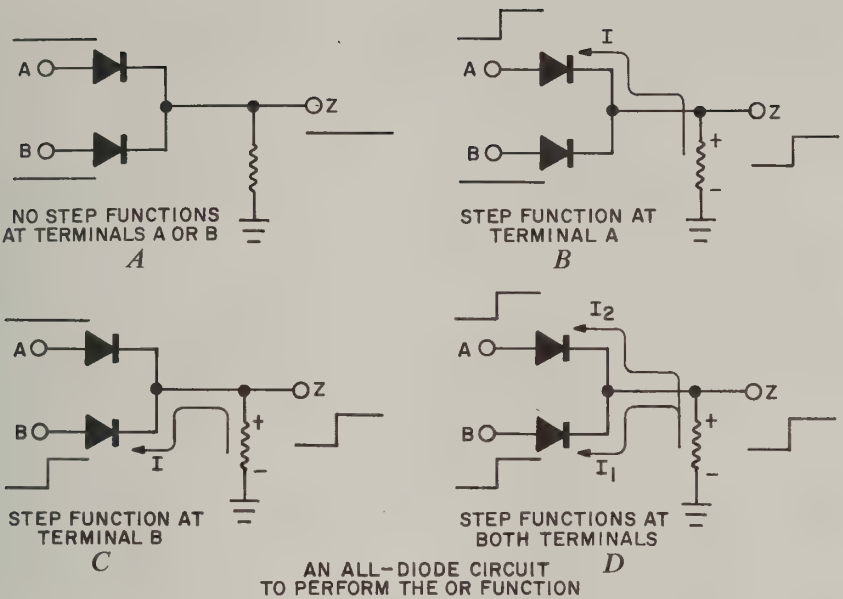


Figure 19

ages in Figure 19 are associated with the logical 1, then the absence of a voltage (or a negative voltage) would indicate a logical 0. In terms of Boolean symbols:

$$\begin{array}{lll} A = 1 & B = 1 & Z = 1 \\ \bar{A} = 0 & \bar{B} = 0 & \bar{Z} = 0 \end{array}$$

Obviously, $\bar{1} = 0$ and $\bar{0} = 1$ since these are the only two digits allowed. Based on these definitions, the digital form of the truth table for the OR function is presented in Table 4. Table 4 indicates the logic level of each input and the resulting output level for each set of input levels. Input points and output points of logical circuits are not normally labeled as such on truth tables.

Figure 20 illustrates a few of the more common logic symbols used to indicate the AND function on a logic diagram. In Boolean algebra, this is indicated by:

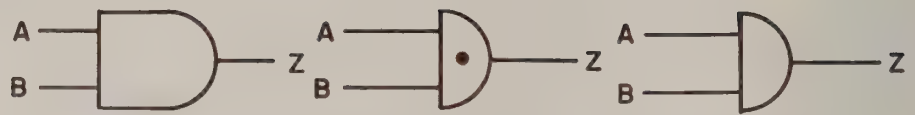
$$A \cdot B = Z$$

implying, that only both conditions (A and B) occurring simultaneously will produce result Z. Figure 21 presents one possible circuit for achieving the AND function. As shown in Figure 21A, both diodes are biased in such a way as to be "on", producing a negative voltage at the output. If this negative voltage is associated with logical 0, then the output must be made less negative in order to produce a logical 1. In Figure 21B, a positive step (from

INPUT POINTS		OUTPUT POINT
A	B	Z
1	1	1
1	0	1
0	1	1
0	0	0

$A + B = Z$
Digital Form of Truth Table for the OR Function

Table 4



SELECTION OF SYMBOLS FOR THE AND FUNCTION

Figure 20

logical 0 to logical 1) is applied to input A, causing the diode in this branch to stop conducting. However, the diode in the other branch continues to conduct, holding the output voltage at the negative (logical 0) level. If the positive step were applied to input B (but not to input A), the branch B diode would go "off", but the branch A diode would stay "on", again maintaining the output at logical 0. Both diodes can be shut "off" by applying simultaneous positive steps to inputs A and B, as shown in Figure 21C. This causes the output voltage to become zero. The output voltage, which is less negative (or more positive) with respect to the logical 0 level, provides a logical 1 indication at the output.

Table 5 is constructed as the truth table for the AND function, using the logical signal level conditions in Figure 21 as a guide. If the input voltage at A or B is negative, there is a negative output voltage indicating a logical 0:

$$0 \cdot 0 = 0.$$

This establishes the first row of the table (shown schematically in Figure 21A). The input at A may be positive going (step to a logical 1) while there is a negative voltage input at B (a logical 0). This, however, leaves the output negative (a logical 0):

$$1 \cdot 0 = 0.$$

This is the second row in the table (shown schematically in Figure 21B). The input at B may be positive going (step to a logical 1) while the input at A is a negative voltage (a logical 0). This also leaves the output negative (a logical 0):

$$0 \cdot 1 = 0.$$

Finally, positive going steps may be applied at A and B simultaneously (both are going to logical 1's). This produces a positive going step output, (a logical 1):

$$1 \cdot 1 = 1.$$

This establishes the last row of the table (shown schematically in Figure 21C). The Boolean symbol summary for the AND function is shown in

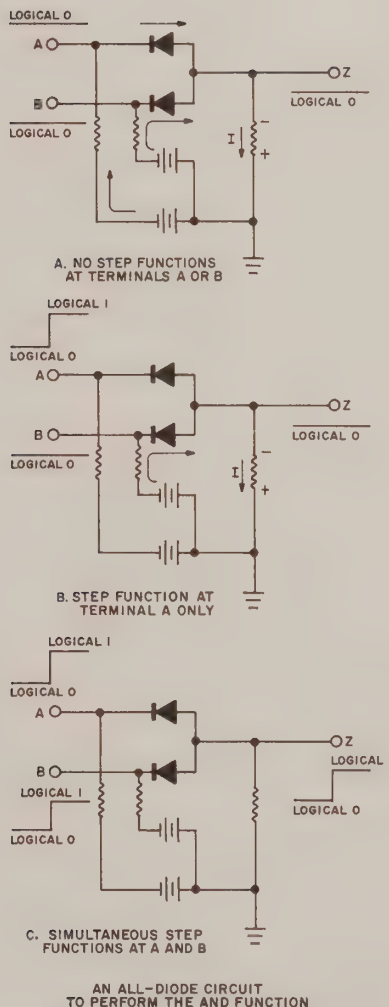


Figure 21

Table 6. The rows in a truth table may be constructed in any desired order, as long as all possible condition combinations are included.

The NOT Function

The NOT function indicates a negative, opposite or inverted condition. Hence, if a positive signal represents a given condition, the absence of the signal (or the application of a negative signal) represents the opposite condition. Manual switches can be used to provide the absence of a signal. They are simply left open until the signal is desired. However, most digital systems are automatic and manual intervention is used only as an "override" capability in the event of a failure or the desire to change the operation in some manner. Furthermore, the full implications of the NOT function are far more subtle than has been described thus far. This is especially true for more complex systems having a multitude of input conditions, arrangements of basic functions in parallel or in series, or—in some cases—more than one output result. For this reason, it is desirable to have a circuit which inverts a signal condition, causing a "true" statement to become "false", and vice versa.

The CE transistor amplifier provides a simple and practical circuit for producing voltage signal inversion. A self-biased NPN arrangement is shown in Figure 22. The forward bias on the base-to-emitter junction allows an output current to flow, establishing a positive output voltage level tending toward the supply voltage. If a positive input signal is chosen to represent A, the increased forward bias on the NPN transistor causes more output current to flow (the transistor tends toward saturation). This, in turn, makes the top of the output resistor less positive, reducing the positive output voltage. Since this is a negative-going output signal (as shown in the figure), it represents $\bar{A}$. If a negative input signal were chosen to represent A, as shown in Figure 22B, the decreased forward bias would cause the top of the output resistor to become more positive, again representing $\bar{A}$. This deliberate use

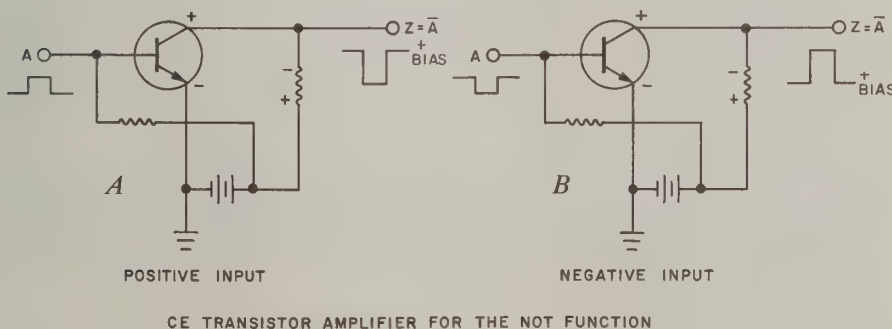


Figure 22

INPUT POINTS		OUTPUT POINT
A	B	Z
0	0	0
1	0	0
0	1	0
1	1	1

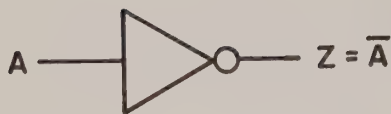
$A \cdot B = Z$
Truth Table for the
AND Function

Table
5

Input	Output
$\bar{A} \cdot \bar{B}$	$\bar{Z}$
$A \cdot \bar{B}$	$\bar{Z}$
$\bar{A} \cdot B$	$\bar{Z}$
$A \cdot B$	Z

Boolean Truth
Table for the
AND Function

Table
6



THE NOT SYMBOL

Figure 23

INPUT POINT	OUTPUT POINT
A	Z
1	0
0	1

Truth Table for the NOT Function

Table 7

of the NOT function is called **COMPLEMENTATION**. It causes a logic circuit to "lie" about the input condition:

$$Z = \bar{A} \quad (\text{NOT}) \quad (7)$$

In terms of the logical digits, the NOT circuit behaves in an identical manner. For an assignment of a positive pulse to represent a logical 1, the output bias voltage level in Figure 22B would represent logical 1. With a positive pulse at the input (logical 1), the decreased positive output voltage would represent logical 0 (Figure 22A):

$$0 = \bar{1}.$$

Also, a decreased positive input voltage would cause a positive signal at the output:

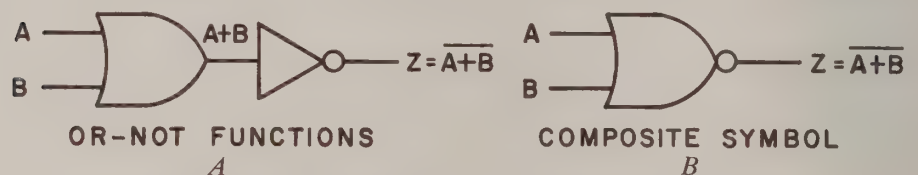
$$1 = \bar{0}.$$

For an assignment of a negative pulse to represent a logical 1, the output bias voltage level in Figure 22A would still be used to represent logical 1. With a negative signal at the input (logical 1), the "increased" positive output voltage would represent logical 0 (Figure 22B). If the circuit used a PNP transistor, bias polarities, signal levels and logical digit assignments would be opposite to that described. However, the complementation operation would not be changed.

Figure 23 illustrates the logic symbol for the NOT function. The triangle indicates amplification and the small circle on the output side indicates signal inversion. The truth table for the NOT function is presented as Table 7.

NOR and NAND Functions

Figure 24A presents a logic diagram representation of a combined OR and NOT circuit. The circuit simply consists of the two basic functions in cascade, to produce what is known as the **NOR** function. This function occurs so frequently in logical systems that it is sometimes classed as a basic function (along with OR, AND and NOT), and is usually treated as a single logic



DEVELOPMENT OF THE NOR FUNCTION

Figure 24

The following Practice Exercise questions cover the subjects which you have just studied. They are:

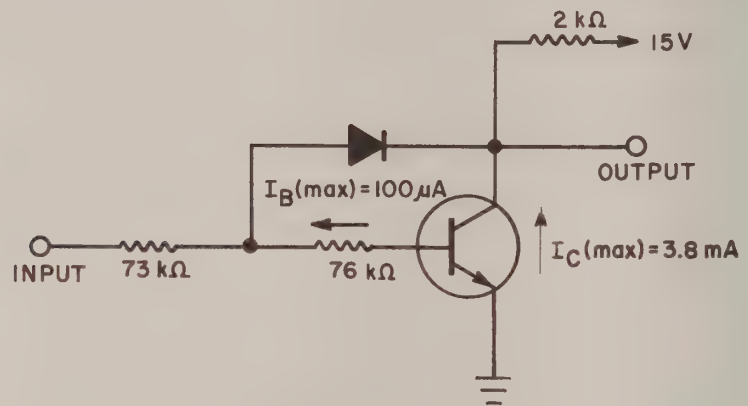
LOGICAL GATING OPERATIONS

BASIC BOOLEAN ALGEBRA

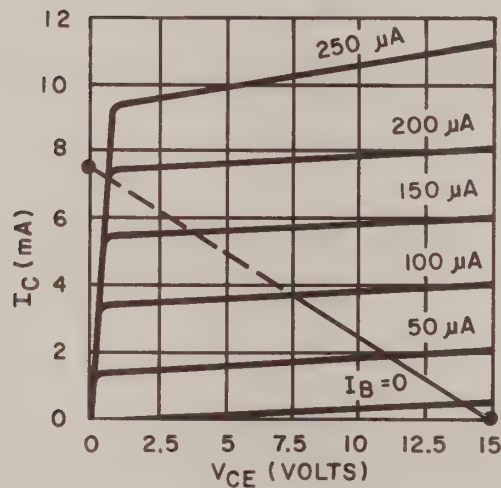
OR AND AND FUNCTIONS

7. What are the three response characteristics normally associated with electronic devices used in pulse applications?
8. Significant storage delay is associated with the junction transistor when it is driven into _____ and then comes out of this condition.
9. What distortions can be expected when transformers are used to couple pulses from one part of a circuit to another?
10. What two distorting components are not always apparent in a circuit schematic?
11. Why is it sometimes desirable to eliminate the saturation condition as one state of operation in transistor switching circuits?
12. The collector voltage clamping technique prevents saturation by holding the _____ above the level normally obtained in a saturated switch.

13. The collector current clamping technique prevents saturation by holding the _____ and, indirectly, the _____ below the levels normally obtained in a saturated switch.
14. What are the upper and lower output voltage levels for the switching circuit shown below?



CIRCUIT



TRANSISTOR CHARACTERISTIC

15. How may the logical decisions, "yes" and "no", be related to the electrical state of a switching device and to the logical digits?
16. In a particular digital system, a positive 5 volt level is used to indicate a logical 1. What voltage level may be used to indicate a logical 0?

17. Can the two logical digits (1 and 0) be related to actual numerical values?
18. What are the three basic logic functions?
19. Using symbols, write "C or D equals X".
20. Using symbols, write "C and D equals X".
21. The AND function implies that the simultaneous application of a logical 1 to each input results in an output logical 1; the OR function implies that the application of the logical 1 to one or more inputs results in an output logical 1. True or False?
22. What additional function is associated with $A + B$, when interpreted as the INCLUSIVE-OR function?
23. Write the complete INCLUSIVE-OR expression for "C or D or C and D equals X".
24. If C is defined by "the door is shut", write the negation symbol for C and its word definition.
25. What is summarized by a truth table?
26. In order to produce the OR function, a circuit must be arranged so that a logical 1 at either (or both) of the inputs will produce logical 1 output. True or False?
27. What is the value of $\bar{1}$?
28. Truth tables using the logical digits do not have the same meaning as truth tables using the Boolean symbols. True or False?

29. How does the truth table for the AND function differ from the truth table of the INCLUSIVE-OR function?
30. The AND function is also referred to as the logical product. Using Table 5, describe how the logical product of two terms differs from the arithmetic product of two terms.

element when shown in a logic diagram. The single-element symbol is shown in Figure 24B. In Boolean algebra, this logic is indicated by a long overbar covering all OR terms:

$$\overline{A + B} = Z \quad (\text{NOR}) \quad (8)$$

which is read, "A or B notted equals Z".

The NOR function may also be indicated by complementing the result, rather than the conditions (A or B equals not Z):

$$A + B = \overline{Z} \quad (\text{NOR}) \quad (8a)$$

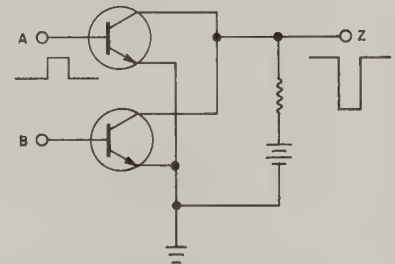
Both of these expressions are equivalent, the selection being one of convenience. This function implies the conversion of a "true" statement into a "false" statement, or vice versa. That is, a positive signal at input A or at input B (possibly representing a logical 1) will produce a less positive signal at the output (to represent a logical 0).

Similarly, a small positive signal at input A or at input B (possibly representing a logical 0) will produce a more positive signal at the output (to represent a logical 1).

Figure 25 shows a NOR circuit using positive logic and consisting of two parallel CE transistor stages feeding a common load. Forward bias is not provided. Hence, when both of the input signals are zero volts (logical 0), the output voltage tends toward the supply voltage (assigned as logical 1). If a positive pulse (logical 1) is applied at input A, the increased current flow will cause the top of the load to become less positive, reducing the positive output voltage and, therefore, producing a logical 0. If a positive signal is applied at input B, a similar action occurs; however, conduction is through the lower transistor. In fact, a positive signal applied simultaneously to both inputs will cause both transistors to conduct, so that a logical 0 output is still produced. Hence, this particular circuit implements the following logic:

$$A + B + A \cdot B = \overline{Z}.$$

This is simply the complementation of the INCLUSIVE-OR function. The truth table for this circuit is provided in Table 8. The truth table is formed on the basis of the following input-output signal conditions: a positive voltage level (logical 1) at both inputs provides an output voltage near zero (logical 0) (first row); a positive voltage level at A and a voltage level near zero at B provides an output voltage level near zero (second row); a positive voltage level at B and a voltage near zero at A provides an output voltage near zero (third row); and the application of a voltage near zero to both inputs provides a large positive voltage level output (fourth row). A compari-



A TRANSISTOR CIRCUIT FOR PERFORMING THE NOR FUNCTION

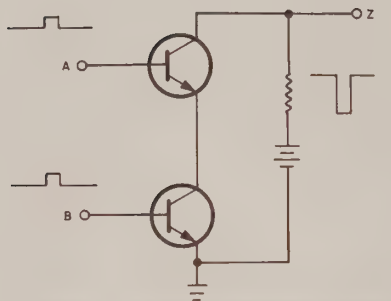
Figure 25

INPUT POINTS		OUTPUT POINT
A	B	Z
1	1	0
1	0	0
0	1	0
0	0	1

$\overline{A + B} = Z$
Truth Table for the NOR Function

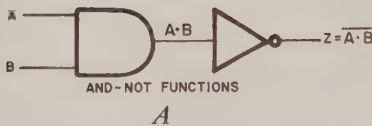
Table 8

CONCEPTS OF SWITCHING LOGIC



A TRANSISTOR CIRCUIT FOR PERFORMING THE NAND FUNCTION

Figure 26



A



COMPOSITE SYMBOL

DEVELOPMENT OF THE NAND FUNCTION

B

Figure 27

INPUT POINTS		OUTPUT POINT
A	B	Z
0	0	1
1	0	1
0	1	1
1	1	0

$$\overline{A \cdot B} = Z$$

Truth Table for the NAND Function

Table 9

son of Tables 4 and 8 will show the total opposition of the OR and NOR functions.

Changing the circuit in Figure 25, so that the collector of one transistor is in series with the emitter of the other transistor, (Figure 26), forces the simultaneous application of logical 1 voltage to both inputs to obtain full conduction through the load (logical 0 output). A single logical voltage at either input (individually) will not permit full current flow, since the alternate series-path transistor will be "off" (logical 1 output). Such an input requirement is indicative of the positive logic AND function. Since the load is fed in a CE manner, output inversion occurs, producing an operation known as the **NAND** function. Like the NOR function, this operation occurs quite frequently in logical systems and is usually treated as a single functional element. The two basic logic symbols are shown in Figure 27A. The usual "composite" symbol is shown in Figure 27B.

In Boolean algebra, the NAND logic is indicated by:

$$\overline{A \cdot B} = Z \quad (\text{NAND}) \quad (9)$$

and is read, "A and B notted equals Z". The result may be complemented, rather than the conditions:

$$A \cdot B = \overline{Z} \quad (\text{NAND}) \quad (9a)$$

Both of these expressions are equivalent. As with the NOR function, the NAND function implies the conversion of a "true" statement into a "false" statement, or vice versa:

$$\begin{aligned} \overline{1 \cdot 1} &= \overline{1} \\ \overline{0 \cdot 0} &= \overline{0} \end{aligned}$$

The truth table for the NAND function is shown in Table 9. The NAND function truth table can be formed by considering the operation of the circuit in Figure 26. Assume that positive voltage inputs (or a large positive output) represent logical 1's and that input voltages near zero (or an output voltage near zero) represent logical 0's. A voltage near zero at both inputs produces a large positive output (first row); a positive voltage level at A and a voltage level near zero at B produces a large positive output (second row); a positive voltage level at B and a voltage level near zero at A produces a large positive output (third row); and positive voltage levels at both inputs produce a voltage level near zero output (fourth row). A comparison of Tables 5 and 9 will show the total opposition of the AND and NAND functions.

The INHIBIT Function

The Boolean logic $A \cdot \bar{B} = Z$ and $\bar{A} \cdot B = Z$ form a part of the possible conditions for the NAND function. However, for the NAND function, Z will occur in the presence of either of these conditions, as well as in the presence of the condition $\bar{A} \cdot \bar{B}$. If a logic circuit is set up so that Z will occur only when $A \cdot \bar{B}$ occurs, or only when $\bar{A} \cdot B$ occurs, the operation is known as the INHIBIT function. The two are distinctly different INHIBIT functions (see Tables 10 and 11). For the function:

$$A \cdot \bar{B} = Z \quad (\text{INHIBIT}) \quad (10)$$

B is the inhibiting condition. That is, if A is logical 1 and B is logical 1, the output is inhibited (Z is logical 0) by $B = 1$. For the function:

$$\bar{A} \cdot B = Z \quad (\text{INHIBIT}) \quad (11)$$

A is the inhibiting condition. That is, if B is logical 1 and A is logical 1, the $A = 1$ condition inhibits the output. Generally, for this function, the simultaneous occurrence of a "true" input at the non-inhibited input and a "false" input at the inhibited input results in a "true" indication; conversely, the simultaneous occurrence of two "true" conditions results in a "false" indication.

Figure 28 illustrates the INHIBIT logic, with a two-button doorbell circuit, for $A \cdot \bar{B} = Z$. One switch, designated A , is normally closed and the other switch, designated B , is normally open. Hence, the condition shown ($A \cdot B$) will not allow the doorbell to ring ($Z = 0$). The condition of B inhibits the condition of A . If we push button A , the doorbell still will not ring ($\bar{A} \cdot B = \bar{Z}$). However, if we push button B , the doorbell will ring ($A \cdot \bar{B} = Z$). For the only other alternative ($\bar{A} \cdot \bar{B}$), where we push both buttons, the doorbell will not ring ($Z = 0$).

Figure 29 provides a combination of diodes and transistors for producing the same INHIBIT function. (A CC stage is used as the output stage so that this stage will not produce any signal inversions.) No forward bias is used; therefore, with no input signals, both transistors are "off", producing only a slight positive output voltage ($A \cdot \bar{B} = \bar{Z}$). When a positive signal is applied at input A , Q_2 is forward biased and D_1 will conduct input loop current. This provides an increased output current and a higher positive voltage at the top of the load, as shown in Figure 29A. Hence: $A \cdot \bar{B} = Z$. In Figure 29B, a positive signal is applied at both inputs. Q_1 inverts the signal, causing a reverse bias on Q_2 (the voltage gain through Q_1 assures a greater negative amplitude than the positive amplitude at input A). For this condition, $A \cdot B = \bar{Z}$ (the signal at B inhibits the action of the signal at A). If there is

A	B	Z
0	0	0
1	0	1
0	1	0
1	1	0

Truth Table for
 $A \cdot \bar{B} = Z$

Table
10

A	B	Z
0	0	0
1	0	0
0	1	1
1	1	0

Truth Table for
 $\bar{A} \cdot B = Z$

Table
11

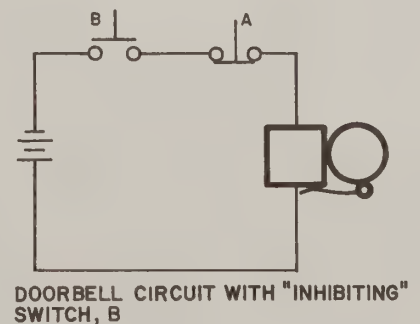


Figure
28

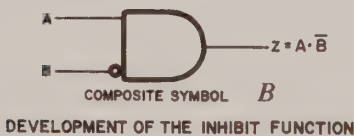


Figure 30

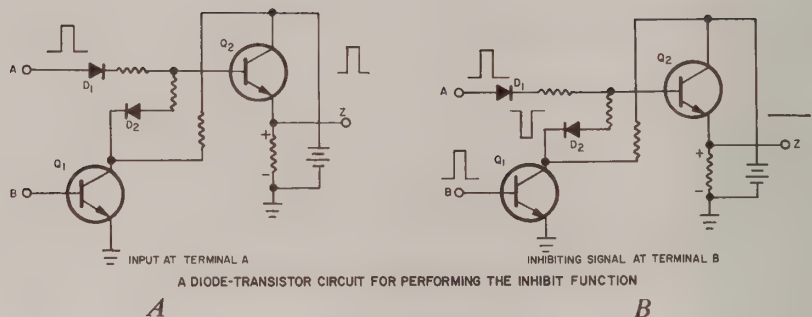


Figure 29

no signal at A and a signal at B, the latter will still be inverted, causing Q_2 to be in cutoff ($\bar{A} \cdot B = \bar{Z}$).

You can see, from the circuit arrangement in Figure 29, that $A \cdot \bar{B} = Z$ is implemented by placing an inverter (NOT) stage between the B input and the switching circuit. This is shown in Figure 30A. The composite logic symbol is shown in Figure 30B. The inhibit circuit along with other logical circuits is often called a gating circuit. In Figure 29, input B is called the gating input. When the input at B has a positive voltage (saturating Q_1), the gate is "off" or "de-energized", and a signal at input A has no effect on the output. However, when the signal at B is less positive (Q_1 is off), the gate is "on" or "energized", allowing signals at A to be reflected in the output.

POSITIVE AND NEGATIVE LOGIC

The switching circuits utilized so far primarily have NPN transistor switches or diodes, biased in such a way that large positive input signals produced conduction; and small positive signals or negative signals produced non-conduction. Such circuitry is known as **POSITIVE LOGIC**. A reversal of the diodes, the diode biasing polarity, or the substitution of PNP transistors (and bias polarity reversal) in any of these logic circuits will produce **NEGATIVE LOGIC**. Negative logic requires a large negative input signal for conduction; and a positive (or small negative) signal for nonconduction. Figure 31 shows the negative logic version of the OR function circuit of Figure 19 and the AND function circuit of Figure 21.

In Figure 31A, a negative signal at either input (or at both inputs) will produce a negative output; resulting in the **INCLUSIVE-OR** operation. In Figure 31B, the positive output (bias) voltage will be made less positive only if a negative signal is applied to both inputs simultaneously, resulting in the **AND** operation. The various signal combinations may be traced for both circuits (assuming a logical 1 for large negative signals and a logical 0 for small negative signals) to produce the respective truth tables. These signal combinations, shown in Tables 12 and 13, are identical to the signal com-

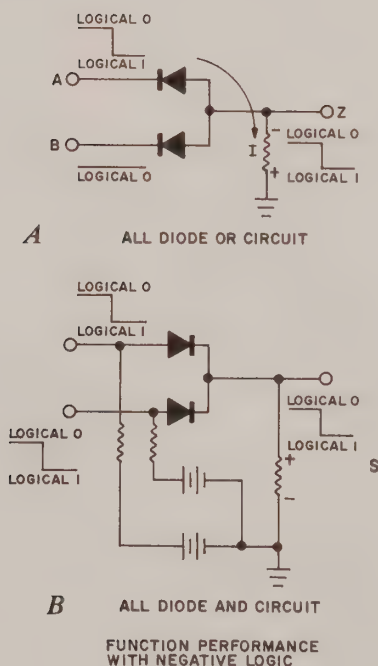


Figure 31

A	B	Z
1	1	1
1	0	1
0	1	1
0	0	0

Truth Table for
the Negative Logic
OR Function

Table 12

A	B	Z
0	0	0
1	0	0
0	1	0
1	1	1

Truth Table for
the Negative Logic
AND Function

Table 13

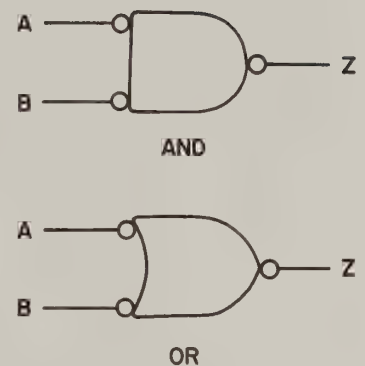
binations of Tables 4 and 5, showing no difference in the basic logical functions. For this reason, there frequently isn't any distinction made between the two on a logic diagram. However, it should always be noted that one or the other is used in order to simplify the troubleshooting and maintenance task. Sometimes the negative logic symbols shown in Figure 32 are used to make this distinction.

"Negative logic" should not be confused with the "logic of negation" (the NOT function and the NOT portion of NOR and NAND functions). The Boolean elements (A, B, Z, etc.) in positive logic use more positive voltage to represent logical 1; the more negative voltage in the negative logic still represents 1. The negated or complementary elements ($\bar{A}$, $\bar{B}$, $\bar{Z}$, etc.), in either case, represent logical 0's.

A negative logic NAND circuit is shown in Figure 33. This NAND circuit is equivalent in operation (and in truth table) to the positive logic NAND circuit in Figure 26. This equivalency can be carried on for all positive and negative logic. The symbols used to indicate negative logic NOR and NAND functions are shown in Figure 34. Notice that all negative logic symbols (Figures 32 and 34) have the small terminal circuit circles on the input side of the symbol. For the basic OR and AND functions, there is also a circle on the output terminal. For the NOR and NAND functions, the small output terminal circle is not present.

MULTIPLE-INPUT REPRESENTATIONS

For all of the logic circuits presented thus far, diodes, resistors, and transistors can be placed in parallel or in series (as needed) so that the number of



SYMBOLS FREQUENTLY USED
FOR NEGATIVE AND AND OR

Figure
32

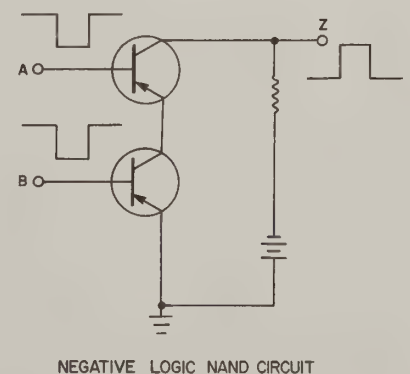


Figure
33



SYMBOLS USED FREQUENTLY
FOR NEGATIVE NAND AND NOR

Figure
34

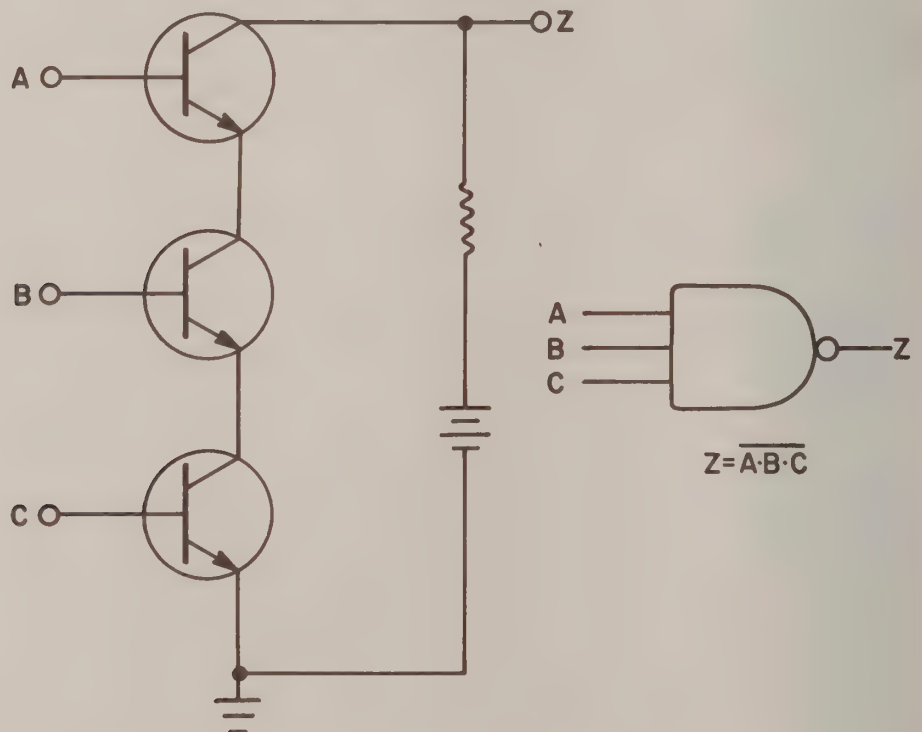
digital inputs is not limited to two. Multiple input OR, AND, NOR and NAND circuits can be formed, as required, to meet specific logic demands.

A positive logic transistor NAND circuit with three inputs is shown in Figure 35. It follows the same design principle as the two-input circuit in Figure 26. However, it must be realized that the addition of each active element usually represents an additional current drain on the supply, and may result in a lower signal output voltage for a given supply value. The logic symbols are adapted to multiple inputs simply by adding input terminal lines. The Boolean expressions are simply expanded to include the appropriate number of OR and AND connectives. For the circuit in Figure 35:

$$\overline{A \cdot B \cdot C} = Z$$

or

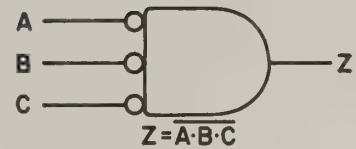
$$A \cdot B \cdot C = \bar{Z}$$



THREE-INPUT, POSITIVE LOGIC NAND CIRCUIT
AND SYMBOL

Figure 35

As always, the Boolean expression applies to both positive and negative logic. If the circuit of Figure 35 used PNP transistors (and the supply polarity were reversed), a negative logic NAND circuit would be produced. It would be distinguished by the logic symbol in Figure 36.



SYMBOL FOR THREE-INPUT, NEGATIVE LOGIC NAND

Figure 36

The truth table for a multiple-input logic circuit may be formed as before: by examining each possible input combination and determining the associated output combination. However, the size of the truth table expands rapidly as the number of inputs increases. The number of columns in the table is equal to the number of inputs plus one (for the output). The number of rows is equal to 2^n , where n is the number of inputs. Because of this, it is easier to develop an understanding of the basic pattern associated with the OR, AND, NOR and NAND functions for two inputs—and then extend this pattern to multiple inputs. These patterns can be derived directly from Tables 4, 5, 8 and 9. They are summarized by the following rules:

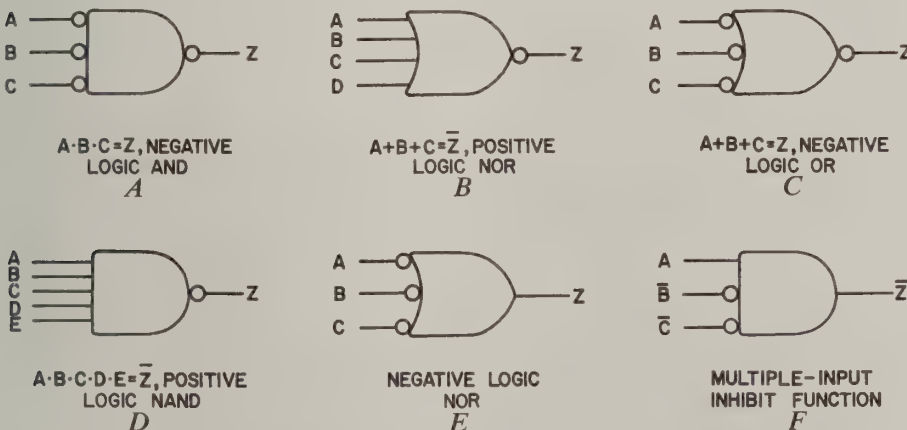
- RULE 1—THE OR FUNCTION OUTPUT IS 0 ONLY WHEN ALL INPUTS ARE 0.
- RULE 2—THE AND FUNCTION OUTPUT IS 1 ONLY WHEN ALL INPUTS ARE 1.
- RULE 3—THE NOR FUNCTION OUTPUT IS 1 ONLY WHEN ALL INPUTS ARE 0.
- RULE 4—THE NAND FUNCTION OUTPUT IS 0 ONLY WHEN ALL INPUTS ARE 1.

A	B	C	Z
1	1	1	0
1	1	0	1
1	0	1	1
1	0	0	1
0	1	0	1
0	0	1	1
0	1	1	1
0	0	0	1

The truth table for the NAND circuit (Figures 35 and 36) is constructed as shown in Table 14. It has four columns (3 inputs + 1 output = 4). It has $2^3 = 8$ rows in order to include every possible combination of input 1's and 0's. All of the input combinations are written, row-by-row, starting with all 1's; then eliminating 1's until all 0's are obtained. Then, using Rule 4, Z is 0 only when all inputs are 1's (first row). The Z-value for all other rows is 1.

Truth Table for 3-Input NAND Circuit

Table 14



EXAMPLES OF MULTIPLE-INPUT LOGIC

Figure 37

CONCEPTS OF SWITCHING LOGIC

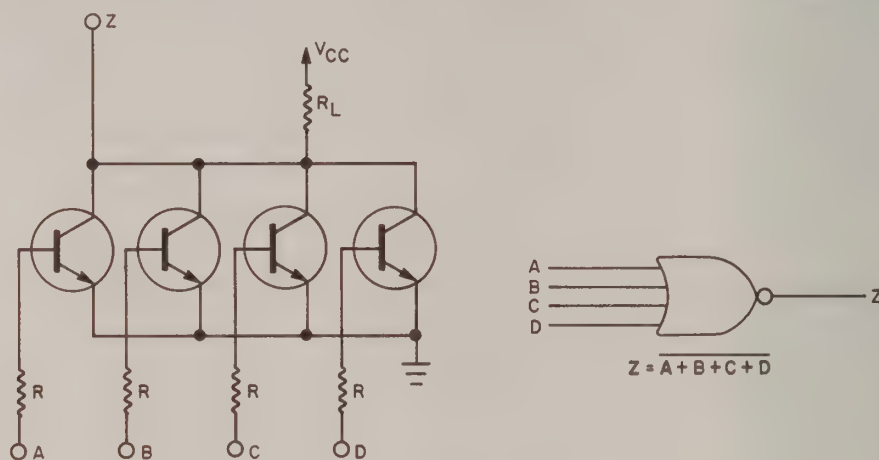
A	B	C	D	Z
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	1	0
0	1	0	1	0
1	0	0	0	0
1	0	0	1	0
1	1	0	0	0
1	0	1	1	0
1	1	1	0	0
1	1	0	1	0
1	0	1	0	0
0	1	1	0	0
1	1	1	1	0

Truth Table for 4-Input
NOR Circuit

Table
15

Six different logic symbols are presented in Figure 37, representing a variety of negative and positive logic elements with different numbers of inputs. The Boolean expression for four of these is also given as a further indication of the fact that all expressions follow the same basic pattern of the two-input expressions. This simple fact can prevent much confusion when more complex operations are described.

The circuit for a four-input, positive logic NOR function is shown in Figure 38. The truth table (see Table 15) has five columns and $2^4 = 16$ rows.



CIRCUIT AND SYMBOL FOR
FOUR-INPUT, POSITIVE LOGIC NOR

Figure 38

SUMMARY

Modern switching systems may utilize hundreds, even thousands, of diodes, transistors, or other solid state (semiconductor) devices. For most applications, these devices will be driven very rapidly from one extreme condition (cutoff) to the opposite extreme conditions (saturation). However, higher speed transistor switching circuits may operate between two states in the active region. Elimination of the saturation condition avoids the storage delay time—a distortion that may increase output pulse width by as much as one microsecond. One method for accomplishing this is to use collector voltage clamping: the use of a diode and clamping bias supply across the transistor output. A second method is the collector current clamping technique: the use of a diode-resistor voltage divider to control base current and (indirectly) collector current. Elimination of the cutoff condition improves switching speed; however, this necessitates the use of complex circuitry.

Logical switching arrangements are based on an association between the two switching states and logical decisions, such as "yes" and "no" or "true" and "false". The states may also be represented by the logical digits, 1 and 0.

There are three basic logical functions: OR, AND and NOT. The OR function produces a logical 1 output when any one of the inputs is a logical 1 (or when all inputs are logical 1's, resulting in a more formal functional definition: INCLUSIVE-OR). The AND function produces a logical 1 output only when all inputs are a logical 1 simultaneously. The NOT function output is an inversion of its input condition. Implementation of the OR and AND functions requires a combination of diodes and/or transistors. The NOT function is implemented by a single CE transistor stage.

NOR and NAND functions result in a set of output conditions exactly opposite to those of the OR and AND functions. They are obtained by following an OR or AND circuit by a NOT circuit. If a NOT circuit is placed between an input terminal and its connection to an AND circuit, the result is an INHIBIT function. This function results in a logical 0 output from the AND function when both inputs are logical 1's or, conversely, produces a logical 1 output only if the inhibiting terminal is a logical 0 when other inputs are logical 1's.

Logic assignments (state voltage levels) may be made in any arbitrary manner, as long as they are used consistently. The two most common, however, are known as positive and negative logic. For positive logic, a logical 1 is associated with a relatively high positive voltage level; a logical 0, with a less positive or negative voltage level. For negative logic, a logical 1 is associated with a high negative voltage level; a logical 0 with a less negative or positive voltage level.

Logical circuits and functions may treat any number of input conditions. The operation, of course, becomes more complicated as more inputs are added. A truth table is used to describe all the possible input combination states and the resulting output states. Such a truth table consists of a column for each input and a column for the output, along with enough rows of entries to consider every possible combination of input states (2^n rows, where n is the number of inputs).

APPENDIX MIL-STD-806B

RELATIVE LOGIC LEVEL MAGNITUDE	POSITIVE LOGIC DIGIT	NEGATIVE LOGIC DIGIT
HIGH	1	0
LOW	0	1

The Following Table of Combinations Illustrates the Applications and Functions of Two Variables and Equivalents

TABLE OF
COMBINATIONS

AND	OR	A	B	X
		H	H	H
		H	L	L
		L	H	L
		L	L	L
		H	H	L
		H	L	L
		L	H	H
		L	L	L
		H	H	L
		H	L	H
		L	H	L
		L	L	L
		H	H	L
		H	L	L
		L	H	H
		L	L	H
		H	H	H
		H	L	H
		L	H	L
		L	L	H
		H	H	L
		H	L	H
		L	H	H
		L	L	H

The following Practice Exercise questions cover the subjects which you have just studied. They are:

LOGICAL GATING OPERATIONS

THE NOT FUNCTION

NOR AND NAND FUNCTIONS

THE INHIBIT FUNCTION

POSITIVE AND NEGATIVE LOGIC

MULTIPLE-INPUT REPRESENTATIONS

31. What purpose is served by the NOT function in terms of input-output signals and in terms of logical definitions?
32. Write an equation showing that X is the complementation of B.
33. Write an equation, using the logical digits, to show the complementation of 1.
34. Draw the single element logic symbol for “C and D equals not X” and write the associated Boolean expression.
35. $\overline{E + F} = Y$ and $E + F = \overline{Y}$ represent the same logical operation. True or False?
36. Why is the circuit in Figure 25 the complementation of the INCLUSIVE-OR function?
37. Write the Boolean expression for “C and D notted equals X”.
38. Using the circuit of Figure 25, assume that positive inputs (and a large positive output) represent logical 0's and that input voltages near zero (or an output voltage level near zero) represent logical 1's. Explain the development of Truth Table 9.
39. What is meant by $A \cdot B = \overline{Z}$?
40. What is the purpose of the INHIBIT function?
41. If $\overline{C}$ is an inhibiting condition for D and the output is X, write the Boolean expression for the INHIBITING function.

42. Using an arrangement similar to that of Figure 29, sketch a circuit to implement the INHIBIT logic, $\bar{A} \cdot B = Z$.
43. Negative logic produces truth tables that are opposite to the positive-logic truth tables for the same function. True or False?
44. How does the basic design of a negative logic circuit differ from that of a positive logic circuit?
45. How does the logical digit assignment differ for positive and negative logic?
46. How many inputs are associated with the Boolean expression: $A + B + C + D + E = X$?
47. What function is performed by the circuit in Practice Exercise 46?
48. How many rows will appear in the truth table for the circuit in Practice Exercise 46?
49. Write the Boolean expression for the symbol in Figure 37E.

IMPORTANT DEFINITIONS

AND—The logical function which produces a logical 1 output only in the simultaneous presence of a logical 1 at all inputs. The operation is also known as the logical product, logical multiplication, conjunction, intersection, and coincidence.

BINARY LOGIC—See DIGITAL LOGIC.

BOOLEAN ALGEBRA—A form of symbolic mathematics used to represent logical statements relating conditions and results. Also known as LOGICAL ALGEBRA.

BUT-NOT—See INHIBIT.

COMPLEMENTATION—See NOT.

DIGITAL ENCODING—The process whereby numerical values are converted into a sequence of logical digits.

DIGITAL LOGIC—Switching circuits, arranged to provide logical operations, by an association of the LOGICAL DIGITS (1 and 0) with the switching circuit input and output signal levels. These signal levels may take on only one of two design values (those of logical 1 and logical 0), rather than being continuously variable. Also known as BINARY LOGIC and SWITCHING LOGIC.

GATING CIRCUIT—A term for a digital logic circuit, having an output and a number of inputs, designed in such a way that the output signal is present only when a certain combination of input signals is present.

INCLUSIVE-OR—See OR.

INHIBIT—The logical function which allows the inhibit input at logical 1 to cancel the effects of all other input conditions resulting in a logical 0 output. However, a logical 0 at the inhibit input allows the signals at the other inputs to control the output. Also known as the "BUT-NOT" operation.

INVERSION—See NOT.

LOGIC—The basic principles and techniques associated with making rational decisions. Also used to refer to the circuits used in implementing digital controls, computers and other operations.

LOGICAL ALGEBRA—See BOOLEAN ALGEBRA.

LOGICAL CONNECTIVE—The functional signs used to indicate AND and OR in BOOLEAN ALGEBRA.

IMPORTANT DEFINITIONS (Continued)

LOGICAL DIGITS—The two digits, 1 and 0, used to indicate the two possible **DIGITAL LOGIC** states associated with switching circuits. Logical 1 and 0 are used to represent “on” and “off”, “yes” and “no”, or “true” and “false” respectively. Also known as **BINARY DIGITS**.

LOGIC DIAGRAM—The **DIGITAL LOGIC** equivalent to a circuit diagram, indicating input and output signal flow between the individual functional units.

LOGICAL FUNCTIONS—The operations associated with digital logic. See **AND**, **OR**, **NOT**, **NAND**, **NOR**, and **INHIBIT**.

LOGICAL GATING—See **LOGICAL SWITCHING**.

LOGICAL SWITCHING—Or **SWITCHING LOGIC**. See **DIGITAL LOGIC**.

NAND—The logical function which allows a logical 1 output for all input conditions except when all inputs are a logical 1. It is the **COMPLEMENTATION**, inversion or **NEGATION** of the **AND** function.

NEGATION—See **NOT**.

NEGATIVE LOGIC—Digital logic designed to represent input conditions and output results with negative voltage levels or negative pulses.

NOR—The logical function which allows a logical 1 output only when all input conditions are logical zero. It is the **COMPLEMENTATION**, inversion or **NEGATION** of the **OR**-function.

NOT—The logical operation which causes the opposite of a given condition. That is, an existing **LOGICAL DIGIT** is converted to the alternate logical digit; an existing switching state is changed to the opposite state (conduction to nonconduction or vice versa); a positive signal is changed to a negative signal, etc. Also known as **COMPLEMENTATION**, inversion and **NEGATION**.

OR—The logical function which allows a logical 1 output result in the presence of any one (or all) of the logical 1 input conditions. When all input conditions (logical 1) can also produce an output logical 1 result, the function is **INCLUSIVE-OR**.

POSITIVE LOGIC—Digital logic designed to represent input conditions and output results with positive voltage levels or positive pulses.

SWITCHING CIRCUIT—See **GATING CIRCUIT**.

IMPORTANT DEFINITIONS (Continued)

SWITCHING LOGIC—See DIGITAL LOGIC.

TRUTH TABLE—A tabulation of all possible combinations of the input conditions, and the associated results, for a digital logic circuit. Entries in the table may be in terms of the logical digits, the symbols from Boolean algebra, or a variety of other techniques (+ and – or H, for high, and L, for low—referring to relative signal levels—are frequently used).

ESSENTIAL SYMBOLS AND EQUATIONS

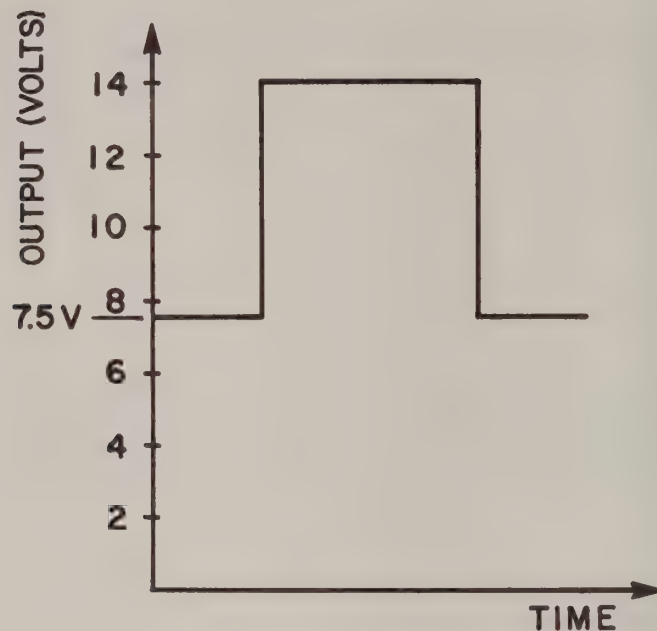
A, B, C, D	Arbitrarily selected input variables for logical switching circuits	
$\overline{A}, \overline{B}, \overline{C} \dots \overline{X}, \overline{Y}, \overline{Z}$	Indication of inversion of the logical switching circuit variables	
I_C'	Collector current at clamped level	
R_L	Load resistor	
V_{CC}	Collector supply voltage	
X, Y, Z	Arbitrarily selected output variables for logical switching circuits	
$+$	Logical connective for OR	
	Logical connective for AND	
$R_2 = \beta \left[\frac{V_{CC}}{I_C'} - R_L \right]$		(1)
$A + B = Z$	(OR)	(2)
$A + B = B + A$		(3)
$A \cdot B = Z$	(AND)	(4)
$AB = Z$	(AND)	(4a)
$A \cdot B = B \cdot A$		(5)
$A + B + A \cdot B = Z$	(INCLUSIVE-OR)	(6)
$Z = \overline{A}$	(NOT)	(7)
$\overline{A + B} = Z$	(NOR)	(8)
$A + B = \overline{Z}$	(NOR)	(8a)
$\overline{A \cdot B} = Z$	(NAND)	(9)
$A \cdot B = \overline{Z}$	(NAND)	(9a)
$A \cdot \overline{B} = Z$	(INHIBIT)	(10)
$\overline{A} \cdot B = Z$	(INHIBIT)	(11)

PRACTICE EXERCISE SOLUTIONS

1. “On” and “off”—These states may also be called “conducting” and “nonconducting”, respectively, although there is always some small degree of current flow associated with the “nonconducting” state.
2. higher; smaller; larger
3. forward; reverse; breakdown
4. (a) 60 millivolts; 22 milliamperes—The peak voltage and current values.
(b) 2.73 ohms— $(60 \times 10^{-3}) / (22 \times 10^{-3})$
(c) 350 millivolts; 3 milliamperes—The valley voltage and current values.
(d) 117 ohms— $(350 \times 10^{-3}) / (3 \times 10^{-3})$
5. supply (assuming no large voltage drop across the collector load resistor due to other current paths)
6. Enhancement mode MOSFET's are used because they can be switched with a single-polarity pulse.
7. Rise time, fall time and propagation delay time.
8. saturation
9. Overshoot, undershoot and ringing.
10. Lead inductance and stray capacitance.
11. Elimination of the saturation condition allows transistor switching at a much higher rate.
12. collector-to-emitter voltage
13. base current, collector current

PRACTICE EXERCISE SOLUTIONS (Continued)

14.



15. "Yes" and "no" may be associated with the "on" and "off" states of a switching device and with the logical 1 and logical 0, as desired for a particular application.
16. Any voltage level less than 5 volts may be used to indicate a logical 0.
17. Yes—A sequence composed of logical 1's and 0's can be used to represent actual numerical values.
18. The three basic logic functions are AND, OR and NOT.
19. $C + D = X$
20. $C \cdot D = X$
21. True
22. As the INCLUSIVE-OR function, $A + B$ includes the simultaneous occurrence of A and B, as well as the occurrence of either A or B so that the complete logical expression becomes: $A + B + (A \cdot B)$
23. $C + D + (C \cdot D) = X$
24. The negation of C is $\bar{C}$, defined by "the door is not shut", or "the door is open".

PRACTICE EXERCISE SOLUTIONS (Continued)

25. A truth table summarizes the complete set of conditions and results for a given logical situation.
26. True
27. $\bar{1} = 0$
28. False
29. The truth table for the AND function has a logical 1 output only when all inputs are a logical 1. The truth table for the OR function has a logical 1 output when any or all of the inputs are a logical 1.
30. The logic product and arithmetic product of two terms (terms being 1's or 0's) are identical.
31. The NOT function causes an inversion of the input signal, which is equivalent to reversing a logical condition.
32. $X = \bar{B}$ or $\bar{B} = X$
33. $\bar{1} = 0$ or $0 = \bar{1}$
- 34.



35. True
36. The circuit in Figure 25 implements the logic: $A + B + A \cdot B = \bar{Z}$ which is the complementation of the INCLUSIVE-OR function: $A + B + A \cdot B = Z$.
37. $\overline{C \cdot D} = X$
38. The first row of truth table 9 will result from the fact that positive voltage levels at both inputs produce an output voltage near zero ($\bar{A} \cdot \bar{B} = Z$ or $\bar{1} \cdot \bar{1} = 1$). The second row results from the fact that a voltage level near zero at A and a positive voltage level at B produce an output voltage near zero ($A \cdot \bar{B} = Z$ or $1 \cdot \bar{1} = 1$). The third row results from the fact that a positive voltage level at A and a voltage level near zero at B produce an output voltage near zero ($\bar{A} \cdot B = Z$ or $\bar{1} \cdot 1 = 1$). The last row results from the fact that voltage levels near zero at both inputs produce a positive output voltage level ($A \cdot B = \bar{Z}$ or $1 \cdot 1 = \bar{1}$).

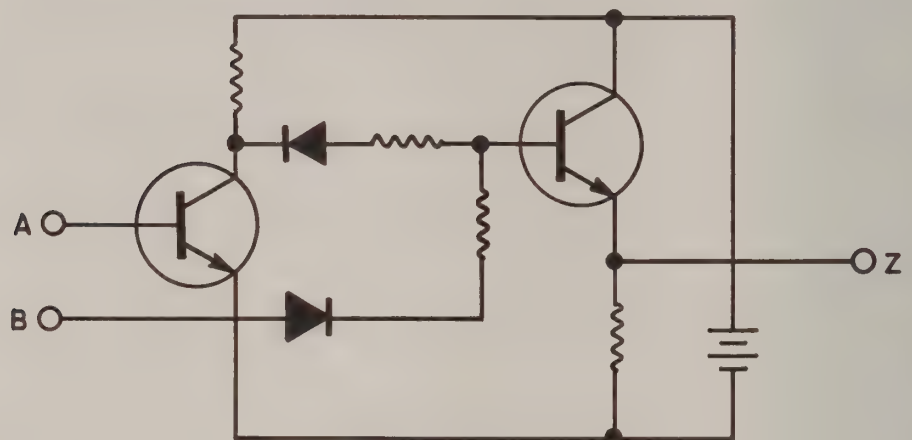
PRACTICE EXERCISE SOLUTIONS (Continued)

39. This indicates that a signal at both inputs produces no signal at the output.

40. It is an inverter; therefore, it inverts the logic level of the inhibit input applied to the gate.

41. $D \cdot \bar{C} = X$

42.



43. False

44. Negative logic circuits have a reversal of diodes, of diode biasing polarity, or the substitution of PNP transistors (and a bias polarity reversal) for NPN transistors.

45. For positive logic, the more positive voltage or less negative voltage is assigned to logical 1. For negative logic, the more negative voltage or less positive voltage is assigned to logical 1.

46. Five.

47. OR

48. $32 - 2^5 = 32.$

49. $\overline{A + B + C} = Z$ (with negative logic)

1. A - A LOGICAL 1 AND 0 -- MUST BE REPRESENTED BY THE OPPOSITE STATES OF AN ACTIVE DEVICE, BUT MAY BE ARBITRARILY ASSIGNED.

Logical algebra is based on the use of a two-state switching circuit.

2. D - FOR A GIVEN TRANSISTOR, THE MOST RAPID SWITCHING CAN BE OBTAINED BY -- ELIMINATING SATURATION AS ONE OF THE SWITCHING STATES.

When a junction transistor is in saturation, both the base-to-emitter and base-to-collector junctions are forward biased, and are acting as "emitters" injecting majority carriers into the base region. This causes the carrier density in the base region to become extremely high. When the transistor comes out of saturation, the excess carriers must flow back into the collector. This process takes an amount of time.

3. D - TRANSISTOR SWITCHES HAVING BOTH STATES IN THE ACTIVE REGION -- DO NOT PROVIDE THE SAME AMOUNT OF OUTPUT VOLTAGE VARIATION AS OTHER TRANSISTOR SWITCHING CIRCUITS.

A greater output voltage variation is obtained by transistor switches being driven from their cutoff region to their active region, or from their active region to their saturation region.

4. B - THE LOGIC LEVELS FOR NPN AND PNP TRANSISTORS -- MUST DIFFER IN VOLTAGE POLARITY.

The basic difference in any NPN or PNP transistor circuit is the source and circuit voltage polarities. For logic circuits the logic levels must also differ in polarity.

5. C - MOST MOSFET SWITCHING CIRCUITS -- USE ENHANCEMENT MODE MOSFET'S.

Enhancement mode MOSFET's can be switched with single-polarity pulses; therefore, they are more commonly used in FET switching circuits. MOSFET's are faster than JFET's and they have a higher input impedance than JFET's. MOSFET's also have a lower input power requirement than JFET's.

6. D - $A + \bar{B} = Z$ -- INDICATES THAT Z WILL BE A LOGICAL 1 EVEN THOUGH B IS A LOGICAL 0.

When B is a logical 0, $\bar{B}$ is a logical 1 and $A + \bar{B} = Z$ can only be a logical 1. Since this expression is for an OR gate, if the input at A and/or B is logical 1, the output at Z will be a logical 1.

7. D - THE NOT FUNCTION -- REVERSES AN INPUT LOGIC CONDITION.

The NOT function indicates a negative, opposite, or inverted logic condition.

8. D - THE NOR AND NAND FUNCTIONS -- HAVE OUTPUT CONDITIONS THAT ARE EXACT OPPOSITES TO THE OUTPUT CONDITIONS FOR OR AND AND, RESPECTIVELY.

For a NOR function, the output is at a logical 0 when any or all of the inputs are a logic 1; for an OR function, the output is at a logical 1 when any or all of the inputs are a logic 1. For a NAND function, the output is at a logical 0 only when all of the inputs are a logic 1; for an AND function the output is at a logical 1 only when all of the inputs are a logical 1.

9. A - POSITIVE AND NEGATIVE LOGIC -- ACCOMPLISH EXACTLY THE SAME FUNCTIONS WHEN USED WITH APPROPRIATE DIGITAL CIRCUIT COMPONENTS.

Boolean elements (A, B, Z, etc.) in positive logic use more positive voltages to represent a logical 1; the more negative voltages in negative logic still represent a logical 1; however, the circuit components and voltages are reversed.

10. D - FOR MULTIPLE-INPUT LOGIC ARRANGEMENTS -- THE NAND FUNCTION IS 0 ONLY WHEN ALL INPUTS ARE 1.

The OR function output is 1 when any or all inputs are 1. The AND function output is 0 when any input is not 1. The NOR function output is 0 when any or all inputs are 1.

QUESTIONS

IMPORTANT—These instructions **MUST** be accurately followed to avoid loss, or errors in grading.

Indicate your answer on this sheet by filling in the box for the most correct answer to each question, as illustrated in the example below.

When all questions have been answered, place the answer card in the proper position to line up the boxes on the card with the boxes on the sheet.

Next, copy the complete lesson code into the space provided on the card, and fill in the answer boxes to correspond with those previously filled in on this sheet.

Before mailing, be certain your correct student number, name and address appear on the card.

LESSON CODE
5205A

Example: VII is an example of a number in which system?

A ☐ (A) Arabic number. (B) Roman numeral. (C) Dewey decimal. (D) Metric.
 B ☒
 C ☐
 D ☐

1. A ☐ A logical 1 and 0 (A) must be represented by the opposite states of an active device, but may be arbitrarily assigned. (B) must be associated with the "on" and "off" states of an active device, respectively.
 B ☐
 C ☐ (C) must be associated with the "off" and "on" states of an active device, respectively. (D) are absolute
 D ☐ numerical values.
2. A ☐ For a given transistor, the most rapid switching can be obtained by (A) providing input pulses of especially
 B ☐ large magnitude. (B) eliminating cutoff as one of the switching states. (C) using both the saturation and
 C ☐ cutoff conditions. (D) eliminating saturation as one of the switching states.
 D ☒
3. A ☐ Transistor switches having both states in the active region (A) are used only to provide negative logic
 B ☐ circuits. (B) cannot be used to represent the binary digits. (C) are primarily low-speed switching circuits.
 C ☐ (D) do not provide the same amount of output voltage variation as other transistor switching circuits.
 D ☒
4. A ☐ The logic levels for NPN and PNP transistors (A) must be represented by the "off" and the "saturated"
 B ☒ conditions. (B) must differ in voltage polarity. (C) can be identical in magnitude and polarity. (D) are
 C ☐ difficult to maintain in terms of absolute value.
 D ☐
5. A ☐ Most MOSFET switching circuits (A) are faster than junction transistor switching circuits. (B) are
 B ☐ slower than JFET switching circuits. (C) use enhancement mode MOSFET's. (D) use depletion mode
 C ☐ MOSFET's.
 D ☐
6. A ☐ $A + \bar{B} = Z$ (A) indicates that Z may be a logical 0 even though B may be a logical 0. (B) indicates that
 B ☐ both A and B must be logical 1 if Z is to be a logical 1. (C) indicates that Z will be a logical 1 whenever
 C ☐ A and B assume opposite states. (D) indicates that Z will be a logical 1 even though B is a logical 0.
 D ☒
7. A ☐ The NOT function (A) does not alter a logical 1 input. (B) serves to amplify a given signal level. (C) must
 B ☐ be accomplished with a common collector transistor arrangement. (D) reverses an input logic condition.
 C ☐
 D ☒
8. A ☐ The NOR and NAND functions (A) must be accomplished with a common base transistor arrangement.
 B ☐ (B) are available in diode form. (C) are exact opposites with the same inputs. (D) have output conditions
 C ☐ that are exact opposites to the output conditions for OR and AND, respectively.
 D ☒
9. A ☐ Positive and negative logic (A) accomplish exactly the same functions when used with appropriate digital
 B ☐ circuit components. (B) produce opposite output truth table conditions for the same functional circuit.
 C ☐ (C) have identical logic level polarities. (D) perform complementary functions when assigned to the
 D ☐ same digital circuit arrangement.
10. A ☐ For multiple-input logic arrangements (A) the OR function output is 1 only when all inputs are 1. (B) the
 B ☐ AND function output is 0 only when all inputs are 1. (C) the NOR function output is 0 only when all
 C ☐ inputs are 0. (D) the NAND function output is 0 only when all inputs are 1.
 D ☒

SWITCHING CHARACTERISTICS

5210

© COPYRIGHT 1973
BELL & HOWELL SCHOOLS, INC.
ALL RIGHTS RESERVED





Electronic counters such as the one shown above use logic switching circuits to do the counting. This particular model will measure frequencies from 5 Hz to 32 MHz and indicate the result by means of a six digit decimal readout.

Courtesy Simpson Electric Co.

CONTENTS

Diode Switching	Page 5
Forward Switching	Page 5
Nonideal Switching	Page 6
Reverse Switching	Page 8
Selection of Diode Types	Page 9
Transistor Switching	Page 10
Common Emitter Switching	Page 10
Switching Time Characteristics	Page 12
Common Collector Switching	Page 15
Common Base Switching	Page 16
Temperature Effects	Page 18
Selection of Transistor Types	Page 19
Vacuum Tube Considerations	Page 20

*Extremes meet and there is
no better example than the
haughtiness of humility.*

—Ralph Waldo Emerson

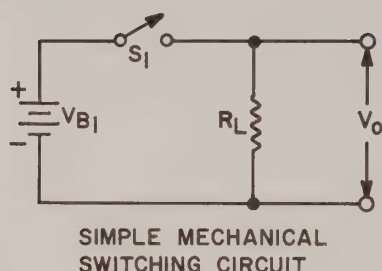


Figure
1

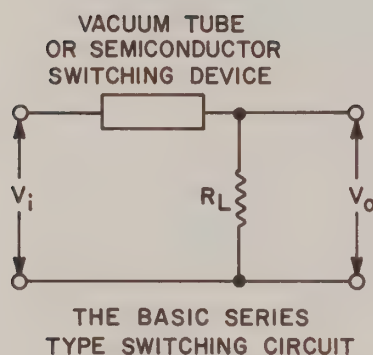


Figure
2

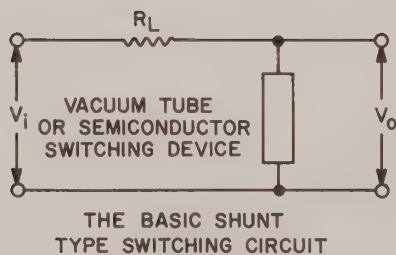


Figure
3

SWITCHING CHARACTERISTICS

A switching circuit has two operating states. This can be best illustrated by examining a simple switching circuit. Figure 1 shows a simple mechanical switching circuit where opening and closing S_1 provides the switching action. The input voltage is supplied by V_{B1} and the output voltage V_o is developed across R_L . The two operating states are usually called the "OFF" and "ON" states or the 0 and 1 states, respectively.

Assume that S_1 is initially open in the circuit of Figure 1, resulting in zero circuit current. With zero current through R_L , V_o is zero. This condition may be called the 0 state. With S_1 closed, V_{B1} develops a circuit current which, in turn, develops a voltage V_o across R_L . This may be called the 1 state.

There are two basic types of switching circuits: series and shunt. A series type switching circuit is shown in Figures 1 and 2. The switching circuit of Figure 2 is in its 1 state (has an output voltage) when the vacuum tube or semiconductor is conducting maximum current for the circuit, thus acting as a closed switch. The circuit is in its 0 state (develops no output voltage) when the vacuum tube or semiconductor is cut off, thus acting as an open switch.

A shunt type switching circuit is shown in Figure 3. To examine the circuit operation, assume that the value and polarity of V_i initially causes the vacuum tube or semiconductor to conduct. The circuit current produces a voltage drop across R_L . However, the output is taken across the vacuum tube or semiconductor, and not across R_L . Since the resistance of the conducting vacuum tube or semiconductor is very low compared to R_L , a small voltage is developed as V_o . As a result, the output is said to be in the 0 state. Notice that this is opposite to the operation of the circuit in Figure 2 where the conduction of the device resulted in a 1 output.

Next, the value and polarity of V_i is reversed, causing the vacuum tube or semiconductor to be cut off. Under these conditions, very little current flows and voltage drop across R_L is insignificant. Hence, almost all of V_i appears at the output terminals as V_o . The output is said to be in its 1 state. Again, we have a condition opposite to that of the circuit in Figure 2.

The internal resistance of the vacuum tube or semiconductor, while it is conducting or cut off, affects the operation of both the series and shunt type switching circuits. An equivalent circuit for both the conducting and cutoff conditions of the series type switching circuit is shown in Figure 4. The resistor R_D replaces the vacuum tube or semiconductor. When the vacuum

tube or semiconductor conducts, the resistance of R_D becomes considerably lower than the resistance of R_L . As a result, very little of V_i is across R_D and V_o is almost equal to V_i . When the vacuum tube or semiconductor switching device is cut off, the resistance of R_D becomes considerably higher than the resistance of R_L . Consequently, almost all of V_i is dropped across R_D and V_o is nearly zero.

Similar operations are shown for the shunt type switching circuit in Figure 5. When the vacuum tube or semiconductor switching device conducts, the resistance of R_D becomes considerably lower than the resistance of R_L . Therefore, most of V_i is dropped across R_L . However, because the output is taken across R_D , V_o is nearly zero. When the switching device is cut off, the resistance of R_D becomes considerably higher than the resistance of R_L . As a result, most of V_i is dropped across R_D . Since the output is taken across R_D , V_o is almost equal to V_i .

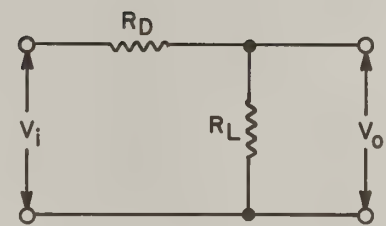
The logic assignments—0 for the less positive or more negative output voltage, and 1 for the more positive or less negative output voltage—are known as positive logic. It is possible to make negative logic assignments: 1 for the more negative or less positive output voltage, and 0 for the more positive or less negative voltage. In practice, positive logic is generally used with positive supply voltage and negative logic is generally used with negative supply voltage. We will use positive logic throughout this lesson, unless otherwise stated.

DIODE SWITCHING

A semiconductor diode can either be conducting (acting as a closed switch) or cut off (acting as an open switch). When conducting, the diode (in a circuit such as Figure 2) develops current through R_L . As a result, a voltage drop V_o is developed across R_L . On the other hand, when the diode is cut off, circuit current is near zero and an insignificant voltage drop is developed across R_L . However, a semiconductor diode is not a perfect switch. It has some low internal resistance when conducting, and some definite high internal resistance when cut off. Thus, the diode does not act as either a perfect short circuit or a perfect open circuit.

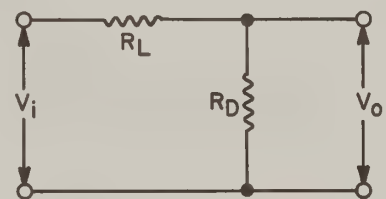
Forward Switching

There are two general methods of utilizing diode characteristics as a switch. The most common method is called **FORWARD SWITCHING**. This method



EQUIVALENT FOR SERIES
TYPE SWITCHING CIRCUIT

Figure
4



EQUIVALENT FOR SHUNT
TYPE SWITCHING CIRCUIT

Figure
5

SWITCHING CHARACTERISTICS

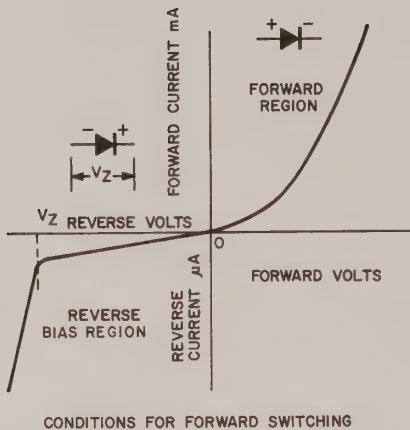


Figure 6

requires the diode to be alternately biased between the forward conduction region and the reverse cutoff region. An example is illustrated in Figure 6. When the voltage applied to a diode makes its cathode negative and its anode positive, the diode operates in the forward bias region. In this region, the circuit resistance and the magnitude of the applied voltage control the current flow. The current flow is relatively large and the diode acts as a closed switch.

When the applied voltage makes the anode negative and the cathode positive, the diode operates in the reverse bias region. In this region, the current flow is controlled by the number of available minority carriers. Since this number is very small compared to the number of majority carriers available in the forward region, the reverse current is relatively small. Hence, the diode resistance is extremely high. The diode resistance and the current flow are almost entirely independent of the applied voltage magnitude. The diode acts as an open switch.

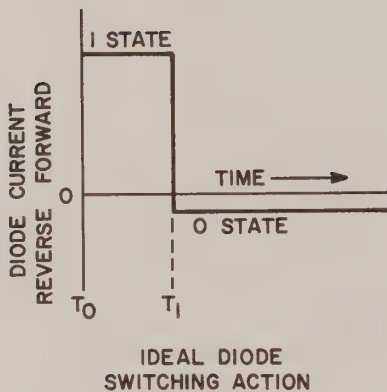


Figure 7

If the reverse bias applied to a diode becomes too high, the diode breaks down and conducts a large reverse current. This point (called the zener level or breakdown level) is shown as V_Z in Figure 6.

The forward switching method generally requires an input voltage of alternating polarity. Furthermore, the reverse input voltage amplitude cannot be allowed to exceed V_Z . The breakdown voltage value depends entirely on the diode construction. It may vary from a few volts to several hundred volts. Large values are usually selected for forward switching applications.

Figure 7 shows the ideal diode switching waveform. Ideally, the diode would start conducting the instant the input voltage rises in the proper polarity, and stop conducting (go into cutoff) the instant the input voltage reverses polarity. The reverse current, of course, is much smaller than the forward current. (The magnitude is exaggerated in Figure 7.)

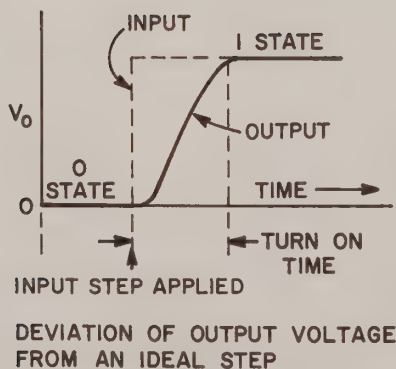


Figure 8

Nonideal Switching

Since no electronic device is ideal, the waveform of Figure 7 cannot be accurately obtained. One nonideal switching characteristic of a diode is the time required for it to change states. The **TURN-ON** time is illustrated in Figure 8. The switching time is a result of two separate conditions: the inherent characteristics of the diode junction and the shunt capacitances in the circuit.

Figure 9 shows the shunt circuit capacitance present in a switching circuit. C_i is the capacitance present across the input terminals. C_o is the capacitance

present across the output terminals. C_i and C_o represent (1) the physical shunt capacitance which is present in the switching circuit, (2) all stray wiring capacitance, and (3) the shunt capacitance introduced into the switching circuit by the semiconductor.

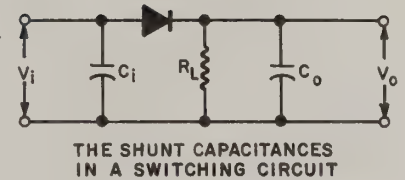


Figure
9

The voltage across a capacitor cannot change abruptly. The rate at which the voltage can change is determined by the value of capacitance and the charging and discharging resistance. Any attempt to change V_o from its 0 state value to its 1 state value must allow time for C_i and C_o to charge through the circuit resistances. Similarly, a change in V_o from its 1 state value to its 0 state value is also delayed because C_i and C_o must discharge through the circuit resistances. In both cases, switching is not instantaneous.

In addition to shunt capacitance effects, semiconductor diodes have an inherent characteristic known as carrier storage, which further increases circuit switching time. When a diode is forward biased, current carriers are introduced into both the P and N-type materials. When the diode is reverse biased, these carriers are temporarily "trapped" in a condition where only minority carriers can flow. Before equilibrium is reached, there is a momentary high reverse current through the diode. This is shown in Figure 10. As shown, the **RECOVERY TIME** interval (T_1 to T_3) is divided into two smaller time intervals. The time interval between T_1 and T_2 is called the **STORAGE TIME** of the diode. During this time interval, the pulse of reverse current through the diode is constant. Its value is governed by the value of reverse bias voltage applied and by the circuit resistance. The time interval between T_2 and T_3 is called the **FALL TIME** of the diode. During this time interval, the large reverse current pulse through the diode drops to its normal low value.

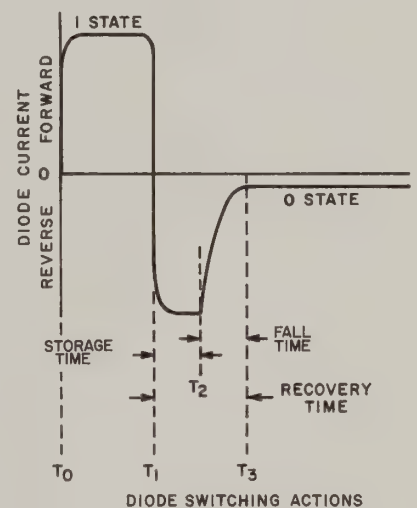
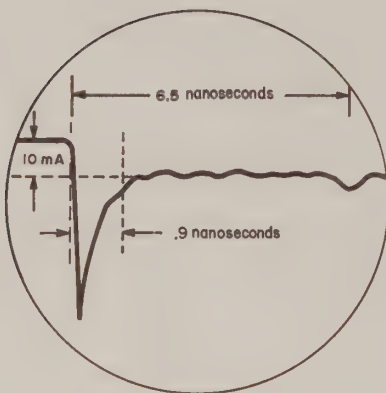


Figure
10

The recovery time and, therefore, the total switching time of a diode is kept small in a number of ways. If the forward bias applied to the diode is low, few current carriers are stored in the N and P-type materials of the diode. Thus, because of their small number, the time required for the current carriers to reach equilibrium is small. One way of keeping the total switching time small is by keeping the resistance in series with the diode high. This reduces forward current and, as a result, reduces the number of stored current carriers. Finally, recovery time may be reduced by forward biasing the diode only for short time intervals. Since the number of current carriers stored is determined by the length of time the diode is forward biased, the recovery time can be kept low by using short forward biasing intervals.

Current carriers are stored only when a semiconductor diode is forward biased. As a result, recovery time effects occur only when a diode is switched from conduction to cutoff. As the device is turned on, a small amount of time is required to inject the current carriers.



OSCILLOSCOPE TRACE OF REVERSE CURRENT PULSE

Figure 11

The capacitance effects already mentioned, together with carrier storage, cause an additional increase in the recovery time of a forward switching circuit. Semiconductor diodes made especially for high speed switching applications have recovery times on the order of a few nanoseconds (10^{-9} seconds). The oscilloscope trace in Figure 11 shows a reverse current pulse that is only 0.9 nanoseconds in duration.

Reverse Switching

The second method applying the diode as a switch is known as **REVERSE SWITCHING**. This method differs entirely from the one described previously, because it uses the breakdown region instead of the forward region, as the ON condition. Special zener diodes are used. These diodes are designed to operate in the breakdown region as long as certain maximum current and power dissipation limits are not exceeded, and therefore, the diode characteristics are not permanently altered. The alternate switching condition is the reverse cutoff region—just as it is in the forward switching circuit.

When the diode is biased to V_Z (refer to Figure 6), its internal resistance drops almost to zero. As a result, the current flow is no longer controlled by the diode. Instead, the external circuit resistance controls the flow. The large reverse current, conducted by a diode in its breakdown region, makes the diode appear as a closed switch. In fact, this region more nearly approximates an ideal switch than does the forward region, since the reverse breakdown resistance may be considerably less than the forward conducting resistance. As stated previously, an ideal switch would have “zero” resistance.

As shown in Figure 12, the input pulse has a negative amplitude of $V_Z + E$. When the diode is in its breakdown region, its voltage drop is V_Z , and the rest of the pulse amplitude, E , appears across R_L with the polarity shown.

DIODE SWITCHING ACTIONS

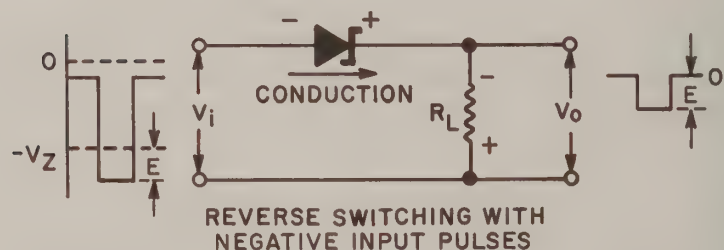


Figure 12

When the applied voltage causes the diode bias to drop below V_Z , operation moves back into the reverse cutoff region. In this region, the diode passes only its normal reverse current, thus acting as an open switch.

The reverse current characteristic of a semiconductor diode may be used in either a series or a shunt-type switching circuit. The advantage of reverse switching is that all changes occur in the reverse biased region. Since this diode operation does not switch from the forward to the reverse regions, there are no carrier storage effects. The recovery time is thereby eliminated. However, only a single polarity switching signal can be used. The polarity depends on diode orientation, as shown in the series-type circuits of Figures 12 and 13. The value of R_L must always be selected to ensure that the maximum zener current is not exceeded.

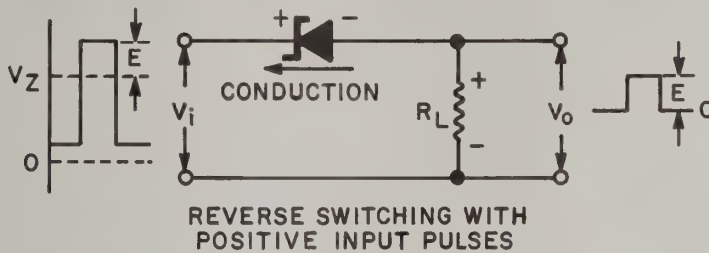


Figure 13

The operation of a shunt-type circuit using reverse switching is illustrated in Figure 14. When the input voltage exceeds V_Z , the diode breaks down and V_o is equal to V_Z . When the diode goes into cutoff, V_o is very nearly equal to the applied voltage.

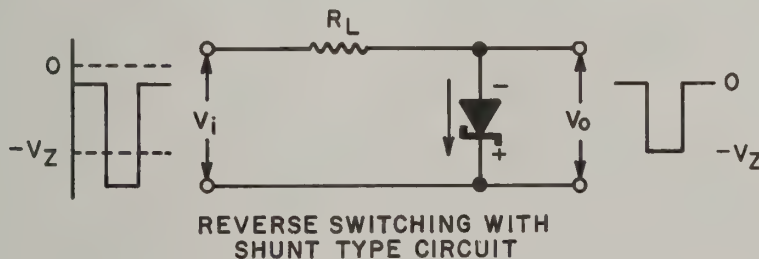


Figure 14

Selection of Diode Types

Semiconductor diodes are made from germanium or silicon. Silicon diodes are used where the normal reverse current must be kept low at high tem-

peratures; they also have a greater power-handling ability than do germanium diodes. However, silicon diodes have a higher forward resistance at all current levels, compared to a similarly doped germanium diode with the same size. The requirements of the switching circuit determine which type of diode is used.

TRANSISTOR SWITCHING

Transistor switching circuits are much more versatile than diode switching circuits, since a transistor is capable of amplifying a voltage and/or current. The output signal developed by a transistor switching circuit can be made considerably larger than the input signal applied to it. This point is very important, since it is often desirable to have an output signal from one switching circuit that is large enough to drive several additional switching circuits.

A transistor acts as an open switch when both its emitter and collector junctions are reverse biased. When the emitter junction is forward biased and the collector junction is either reverse biased or forward biased, a transistor acts as a closed switch, in the active and saturated regions respectively. A transistor is not a perfect switch, however. It is a switch in the respect that its internal resistance is low when it is "closed" (conducting) and considerably higher when it is "open" (cut off). Their switch characteristics, along with their amplification properties, make transistors suited for use in switching applications.

As with diodes, transistor switching circuits may be of either the series or shunt-type. In a series-type switching circuit, a voltage appears across the load resistor when the transistor conducts. In a shunt-type switching circuit, the output voltage appears across the transistor output terminals when the device is cut off. Conventional transistor amplifiers can be connected in any one of three circuit configurations: common emitter, common base, or common collector. Likewise, transistor switching circuits may use any of these configurations. Each configuration has certain advantages and disadvantages.

Common Emitter Switching

The most widely used transistor switching circuit configuration is the common emitter. This wide usage is largely because the common emitter configuration provides the highest power gain, and thus delivers a large output signal.

The following Practice Exercise questions cover the subjects which you have just studied. They are:

DIODE SWITCHING

FORWARD SWITCHING

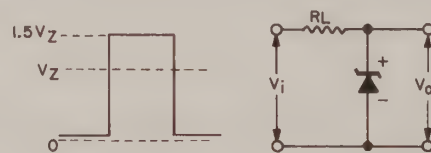
NONIDEAL SWITCHING

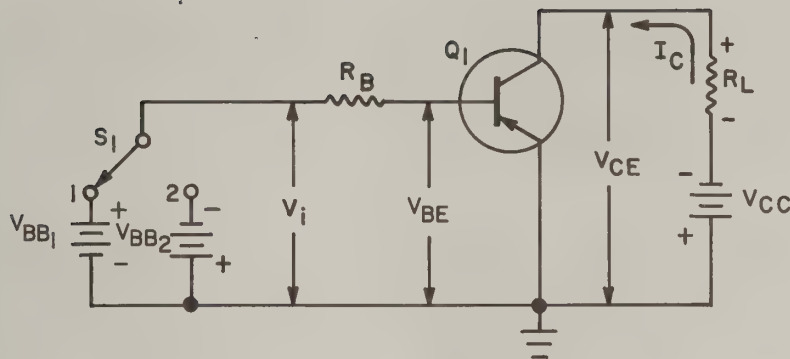
REVERSE SWITCHING

SELECTION OF DIODE TYPES

1. Figure 1 is a (series, shunt) type of switching circuit.
2. A switching circuit has _____ operating states.
3. A diode is a perfect switch in the respect that, when conducting, its internal resistance is zero and, when cut off, its internal resistance is infinite. True or False?
4. In a series-type switching circuit, an output signal is developed when the diode is (a) conducting, (b) cut off.
5. In a shunt-type switching circuit, an output signal is developed when the diode is (a) conducting, (b) cut off.
6. A diode acts as an open switch when operated in its forward region. True or False?
7. If the reverse bias applied to a semiconductor diode is increased to a high enough value, the diode breaks down and can conduct a large reverse current. True or False?
8. If a logical 1 is +5 volts and a logical 0 is +1 volt, positive logic is being used. True or False?
9. If -10 volts is a logical 1 and +1 volt is a logical zero, the type of logic being used is (a) positive, (b) negative.
10. A circuit using +4 volts as a logical 0 and +2 volts as a logical 1 would be using positive logic. True or False?
11. Circuit capacitance prevents a switching circuit from changing states instantaneously. True or False?
12. Current carriers are stored in a semiconductor diode when the diode is (a) forward biased, (b) reverse biased.

13. What is the cause of the large pulse of reverse current through a semiconductor diode when it is switched from forward to reverse bias?
14. If the number of current carriers stored in a semiconductor diode is kept low, the diode recovery time will be short. True or False?
15. For the series-type switching circuit with positive logic, the output is in the 0 state when the diode is conducting and in the 1 state when the diode is cut off, regardless of whether forward or reverse switching is used. True or False?
16. What two restrictions are placed on the applied voltage for a diode reverse switching circuit?
17. In the figure shown below, what does the output look like?





BASIC COMMON EMITTER SWITCHING CIRCUIT

Figure 15

Figure 15 shows a basic common emitter shunt-type switching circuit employing a PNP transistor. An NPN transistor can also be used if the polarities of the bias and collector supply batteries in the circuit are reversed. Batteries V_{BB1} and V_{BB2} , together with S_1 , provide an input voltage V_i for the circuit. V_{CC} supplies reverse bias to the collector junction of Q_1 . R_B represents any resistance present in the base circuit of Q_1 , limiting base current. R_L is the collector load resistor.

V_{BE} is the voltage between the base and emitter of Q_1 , while V_{CE} is the voltage between the collector and emitter of Q_1 . The output signal V_o is taken from the collector of Q_1 to ground. (In this case V_o is the same as V_{CE} .)

The operation of the common emitter switching circuit shown in Figure 15 may be studied by examining the characteristic curves of Figure 16. With R_L equal to 530Ω and V_{CC} (Figure 15) equal to $-9V$, the load line of Figure 16 is drawn as shown. Changes in base current cause the operating point to move up or down along the load line. For example, zero base current causes the collector current and voltage values indicated by point A, while $80\mu A$ of base current causes the values of collector current and voltage corresponding to point C.

The characteristic curves of Figure 16 have been divided into three regions: cutoff, active and saturation. The cutoff and saturation regions are shaded and the active region is unshaded.

Assume that S_1 in Figure 15 is initially in position 1. V_{BB1} makes the base of Q_1 positive with respect to its emitter, thus reverse biasing the emitter junction of Q_1 . With reverse bias applied to the Q_1 emitter junction, base current, I_B , is zero. With I_B equal to 0, collector current I_C is at the very low value shown as point A in Figure 16. With I_C at a very low value, the voltage

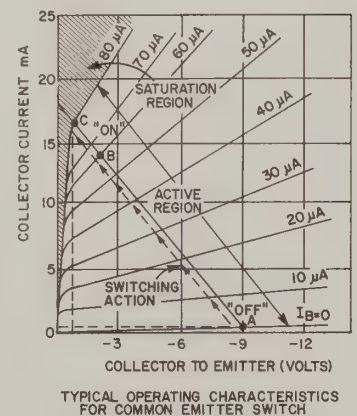


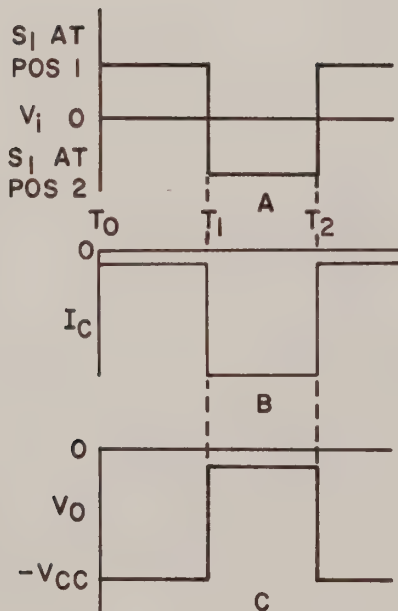
Figure 16

across R_L is very small. Therefore the voltage, V_{CE} , from the collector of Q_1 to ground is almost equal to V_{CC} . Figure 16 illustrates this since point A represents a collector voltage of almost -9 volts. Because output voltage V_o is almost equal to V_{CC} , the circuit is in its 1 state (note the negative logic), developing an output signal.

Now assume that S_1 is placed in position 2. V_{BB_2} makes the base of Q_1 negative with respect to its emitter, thus forward biasing the emitter junction. Due to the resulting base current, I_B , the current gain of Q_1 produces a large collector current, I_C . The voltage across R_L , due to I_C , subtracts from V_{CC} , lowering the Q_1 collector-to-ground voltage, V_{CE} .

If the base current of Q_1 is $80 \mu A$, the values of collector voltage and current correspond to those at point C in Figure 16. Notice that the collector voltage at point C is less than a volt. V_o (which is equal to V_{CE} in this circuit) has decreased and the circuit is now in its 0 state. Since both base and collector currents are flowing, the collector and emitter junctions both have low internal resistances and act as closed switches.

The circuit of Figure 15 develops a large output signal when Q_1 is nonconductive. Conversely, the same circuit develops a very low output signal when Q_1 conducts. Thus, this circuit is a shunt switch arrangement similar to that of Figure 3. Due to the amplification provided by Q_1 , only a very small input signal is needed to control a large output signal.



IDEAL TRANSISTOR SWITCHING WAVEFORMS

Figure 17

Since Q_1 can act as a closed switch in either its active or saturation region, two modes of operation are possible. If Q_1 is operated in its cutoff or active regions only (points A and B), it is called an unsaturated switching circuit. Q_1 may also be operated as a saturated switch by biasing it to operate as a closed switch at point C, in the saturated region, rather than at point B.

Saturated switching has the advantage of low transistor power dissipation in both of its operating states. However, a saturated switching circuit has a longer switching time than an unsaturated circuit. Even though an unsaturated circuit dissipates more power, the smaller switching time is a great advantage in many applications.

Switching Time Characteristics

The time-dependent characteristics of the circuit of Figure 15 may be examined by rapidly switching S_1 between positions 1 and 2 and observing the waveshapes of I_C and V_o . Rapidly switching S_1 between positions 1 and 2 produces an input voltage V_i with the waveshape shown in Figure 17. In

Figure 17A, between times T_0 and T_1 , switch S_1 is in position 1 and V_i is positive. At T_1 , S_1 is switched to position 2 and V_i immediately becomes negative. It remains negative until T_2 . At T_2 , S_1 is switched back to position 1 and V_i immediately becomes positive. The waveshape of Figure 17A could be generated by a multivibrator, rather than by the battery-switch arrangement.

If we have an ideal system in which the switching time is zero, the switching circuit of Figure 15 would change states the instant the input voltage changes levels. Between T_0 and T_1 , when V_i is positive, Q_1 is reverse biased and, thus, cut off. As a result, I_C , shown in Figure 17B, is at its low cutoff value. With I_C near zero, the voltage across R_L is very small and the output voltage across Q_1 is almost equal to $-V_{CC}$, as shown in Figure 17C. At time T_1 , when S_1 is switched from position 1 to position 2, V_i immediately becomes negative. Ideally, Q_1 would immediately become forward biased and, therefore, conductive. As a result, at T_1 , collector current I_C immediately increases and output voltage V_o across Q_1 immediately decreases. As long as S_1 remains in position 2, I_C is maximum, V_o is nearly zero, and the circuit is in its 0 state (using negative logic). At T_2 when S_1 is switched from 2 back to position 1, V_i immediately becomes positive and (ideally) I_C immediately decreases. As a result, a T_2 output voltage V_o immediately increases and the circuit is now in its 1 state. Ideally, therefore, no time is required for the circuit to change state.

The ideal collector current and output voltage waveshapes shown in Figures 17B and 17C are not obtained in actual switching circuits. Circuit capacitances and inherent characteristics of semiconductors prevent switching circuits from changing states instantaneously. With the input voltage waveshape of Figure 17A (repeated as Figure 18A), the output voltage would have the waveshape shown in Figure 18C.

Notice that at time T_1 , when S_1 is switched from position 1 to position 2, V_o does not immediately decrease, as in Figure 17C. Also, collector current I_C , as shown in Figure 18B, does not immediately increase. A definite amount of time is required for output V_o to change from its 1 state value to its 0 state value, and also for collector current I_C to change from near zero to its maximum value.

In transistor switching circuits, turn-on time is usually defined in terms of collector current. It is the time required for collector current to increase from its cutoff value to 0.9 or 90% of its maximum value. Turn-on time is shown in Figure 18C.

Turn-on time is a result of a number of actions. At T_1 , V_i changes from positive to negative and the bias of Q_1 (Figure 15) is changed from reverse to

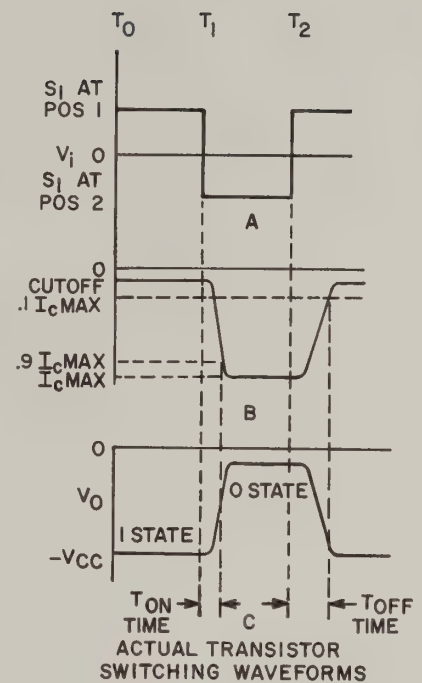


Figure 18

forward. However, the base-emitter capacitance of Q_1 prevents the emitter junction of Q_1 from becoming immediately forward biased. Some time is required for the base-emitter capacitance of Q_1 to discharge from the value of V_1 , between T_0 and T_1 , and charge to the value of V_1 after T_2 .

A second reason for turn-on time is that it takes some time for the current carriers in the transistor to start moving. Collector current starts to increase a short time after the emitter junction becomes forward biased.

The largest cause of turn-on time delay is circuit (possibly stray) capacitances, which limit the speed at which V_o can change. Even though I_C may have reached its maximum value, some time is required for V_o to change because circuit capacitances must charge and discharge.

At time T_2 , V_1 changes from negative to positive and the bias on Q_1 changes from forward to reverse. As shown in Figure 18C, V_o does not increase immediately at T_2 to its 1 state value as in Figure 17C. The output voltage stays at its 0 state value for some time after T_2 , indicating that a definite time is required for V_o to change from its 0 state value to 1 state value. Likewise, for I_C to decrease from maximum to its cutoff value requires some time.

Turn-off time is usually defined in terms of collector current, as the time required for current to change from its maximum value to 10% of its maximum value. Turn-off time is also shown in Figure 18C.

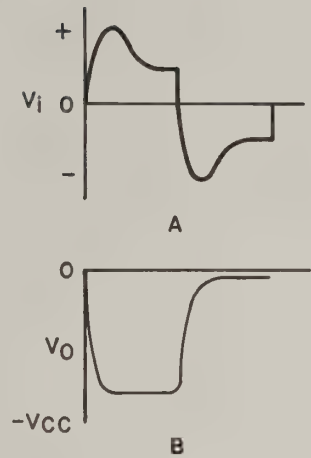
The turn-off time is caused primarily by carrier storage in the transistor. Between T_1 and T_2 , Q_1 in Figure 15 is forward biased. While Q_1 is forward biased, current carriers are being introduced or stored in the base region of Q_1 . At T_2 , when Q_1 is reverse biased, the current carriers that were previously stored in the base region are swept out. As a result, base current I_B flows. The current gain of Q_1 provides an amplified version of this current in the form of collector current I_C . The base current and resultant collector current exist until all the current carriers have been swept out of the base region of Q_1 . In conjunction with this action, I_C decreases to its cutoff value and V_o increases to its 1 state value. The output voltage remains at this value and the circuit remains in its 1 state until V_1 changes. As with the turn-on time, circuit capacitances also contribute to the turn-off time.

Since the turn-on time of a transistor switching circuit is determined primarily by circuit capacitance, by keeping circuit capacitances small, a small, turn-on time can be obtained. If further reductions in turn-on time are necessary, overdriving may be used. Overdriving refers to the use of larger than necessary input signals to change the state of a switching circuit. As a result, the switching circuit starts to change states rapidly; then, once the change

is partially completed, the input signal is reduced to a value just large enough to maintain the desired operating state. The input signal waveform of Figure 19A is an overdriving pulse.

The total turn-off time is determined primarily by storage time. This is the time required for the current carriers stored in the base region of the transistor to be swept out. It is greater in a saturated switching circuit than in an unsaturated one because more current carriers are stored in the base region of a saturated transistor. Storage time may be reduced by overdriving. The large pulse of reverse bias sweeps the stored carriers out faster and, as a result, turn-off time is reduced.

Note, in Figures 17, 18 and 19, that the polarities of the input voltage V_i and the output voltage V_o are opposite. When the input signal is positive, the output signal is maximum negative. On the other hand, when the input signal is negative, the output signal is less negative or more positive. A polarity reversal between V_i and V_o occurs only in the common emitter configuration. This factor must be considered when a common emitter switching circuit is performing logical operations.



DECREASED SWITCHING TIMES WITH OVERDRIVING INPUT

Figure 19

Common Collector Switching

To examine the switching characteristics of the circuit of Figure 20, assume that S_1 is initially at position 1. Notice that Q_1 is an NPN transistor. With S_1 at position 1, V_{BB1} makes the base of Q_1 negative with respect to the emitter, thus reverse biasing the emitter junction. Under these conditions, base current I_B is zero. Battery V_{CC} reverse biases the collector junction of Q_1 and, as a result of both of these biases, collector current I_C is zero. Therefore, both the emitter and collector junction resistances of Q_1 are high and they act as open switches. Since both I_B and I_C are zero, no current exists through R_L in the emitter circuit. As a result, the voltage across R_L (output voltage V_o) is zero. Thus, with S_1 in position 1, the circuit is in its 0 state (positive logic).

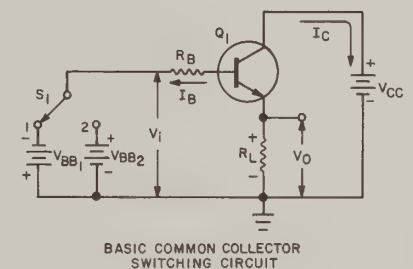


Figure 20

The common collector switching circuit shown in Figure 20 has a cutoff region similar to the common emitter switching circuit (see Figure 16). The common collector circuit reaches a "saturation" condition whenever the base voltage value exceeds the collector voltage value. For this condition, the collector junction will not allow the base voltage to exceed the collector voltage by more than a diode voltage drop.

Now if S_1 is moved to point 2, V_{BB2} makes the base of Q_1 positive with respect to the emitter, forward biasing the emitter junction. As explained earlier, some time is required for the base-emitter junction capacitance of a

transistor to discharge to zero and charge up to the new bias voltage. This action, together with the time required for current carriers in the transistor to start moving, prevents Q_1 collector current I_C from immediately increasing. I_C reaches some maximum value, which is determined by the value of I_B , R_L and V_{CC} .

Collector current I_C through R_L develops an output voltage and the circuit is now in its 1 state (positive logic). The circuit remains in this state, delivering an output signal, until V_i is changed by moving S_1 .

Suppose S_1 is quickly switched back to position 1. V_{BB_1} now applies reverse bias to the emitter junction of Q_1 . However, due to the storage effects of Q_1 , I_C does not immediately decrease. When I_C drops to zero, the voltage across R_L and output voltage V_o drops to zero and the circuit is back in its 0 state. It remains in this state until V_i is changed.

The common collector switching circuit of Figure 20 corresponds to the series switch arrangement of Figure 4. Q_1 in Figure 20 must be acting as a closed switch before an output is developed. When Q_1 is cut off, acting as an open switch, V_o is zero.

Like the common emitter switching circuit of Figure 15, the common collector switching circuit of Figure 20 also acts as an impedance-matching device. Its input impedance is typically around $350\text{ k}\Omega$ and its output impedance is typically around 500Ω . This makes it suitable for matching a high output impedance circuit to a low input impedance circuit.

Unlike the common emitter circuit, the common collector circuit does not produce any polarity reversal between V_i and V_o . When S_1 is in position 1 (Figure 20), V_i is negative with respect to ground and Q_1 is cut off. As a result, V_o is zero. On the other hand, when S_1 is at position 2, V_i is positive with respect to ground and Q_1 conducts collector current I_C . Collector current through R_L produces a voltage of the polarity indicated (positive with respect to ground). Therefore, when V_i is positive, V_o is positive, and when V_i is negative, V_o is zero. The common collector circuit does not produce any voltage gain. That is, the output V_o will not be greater than V_i . The common collector circuit does have current gain, however, and consequently power gain is closely equal to current gain.

Common Base Switching

The third transistor switching arrangement is the common base configuration. It is not as widely used as the common emitter or common collector

The following Practice Exercise questions cover the subjects which you have just studied. They are:

TRANSISTOR SWITCHING

COMMON EMITTER SWITCHING

SWITCHING TIME CHARACTERISTICS

COMMON COLLECTOR SWITCHING

18. In addition to its function as a switch, a transistor provides amplification. True or False?
19. In what three circuit configurations may a transistor switching circuit be connected?
20. When a transistor is operating in its active or saturated regions, its internal resistance is _____ and it is acting as a (an) _____ switch.
21. When a transistor is operating in its cutoff region, its internal resistance is _____ and it is acting as a (an) _____ switch.
22. A transistor operated as a saturated switch dissipates more power than one operating as an unsaturated switch. True or False?
23. The common emitter switching circuit of Figure 15 acts as a (a) series-type switch, (b) shunt-type switch.
24. A certain amount of time is required for a transistor to change from conduction to cutoff. The time delay is primarily the result of (a) circuit capacitance, (b) current carrier storage, (c) electron transit time.
25. What is the predominant characteristic contributing to turn-on time?
26. Both turn-on and turn-off times can be reduced by _____ the switching circuit.
27. Regardless of the type of switching configuration, a consistent definition of turn-on and turn-off time is obtained by using the _____ as the reference waveform.
28. The common emitter switching circuit develops a polarity reversal between its input and output voltage signals. True or False?

29. Unlike the common emitter switch, the output voltage of the common collector switch is taken across the _____ causing it to operate as a _____-type switch.
30. A common collector switching circuit produces a polarity reversal between its input and output signals. True or False?
31. The input impedance of the common collector switching circuit is _____ and its output impedance is _____.

configurations. Figure 21 shows a common base switching circuit employing a PNP transistor. Q_1 in Figure 21 may be operated as a saturated or unsaturated switch. Also, the actions which prevent the circuit of Figure 21 from instantaneously changing operating states are the same as those described for the common emitter circuit of Figure 15.

To examine the operation of the circuit of Figure 21, assume that S_1 is initially in position 1. At this setting, V_{BB1} makes the emitter of Q_1 positive with respect to its base, forward biasing the emitter junction. Under these conditions, base current flows. Due to the current gain of Q_1 , I_C is an amplified version of I_B . You should recall that the transistor itself has current gain; however, the common base configuration has current gain that is a little less than unity since the input current, I_E , is slightly more than the output current, I_C . Collector current through R_L develops a voltage across R_L with the polarity shown. The voltage across R_L subtracts from V_{CC} and reduces the collector-to-base voltage, V_{CB} , of Q_1 . Since V_o is approximately V_{CB} , V_o is also low. Thus, with S_1 (Figure 21) at position 1, V_o is minimum and the circuit is in its 0 state (negative logic).

If S_1 is switched to position 2, V_{BB2} makes the emitter of Q_1 negative with respect to its base, reverse biasing the emitter junction. This tends to reduce I_B and I_C to zero. As mentioned earlier, I_B and I_C do not immediately drop to zero. This time delay is determined by the storage time and fall time effects of Q_1 . After the delay has passed, I_C and I_B become zero. With zero I_C , the voltage across R_L is zero. As a result, V_{CB} (V_o) is equal to V_{CC} . Therefore, with S_1 in position 2, V_o is maximum and the circuit is in its 1 state. The circuit remains in this state until S_1 is switched to its alternate position.

If S_1 is switched back to position 1, the emitter junction of Q_1 becomes forward biased. However, I_B and I_C do not immediately increase, due to the charge and discharge time of the base-emitter junction capacitance and the time required for current carriers to start moving. After this time delay has passed, I_B and I_C reach their maximum levels. As a result, the voltage across R_L becomes large and V_o decreases. Now the circuit is back in its 0 state (minimum output voltage).

The common base switching circuit, like the common emitter switching circuit, acts similar to the shunt switch arrangement of Figure 3. Q_1 in Figure 21, for negative logic, must be nonconductive (acting as an open switch) before an output signal is developed; and conductive (acting as a closed switch) before V_o can be zero.

The common base switching circuit, like the other two transistor circuits described in this lesson, can act as an impedance-matching device. Its input

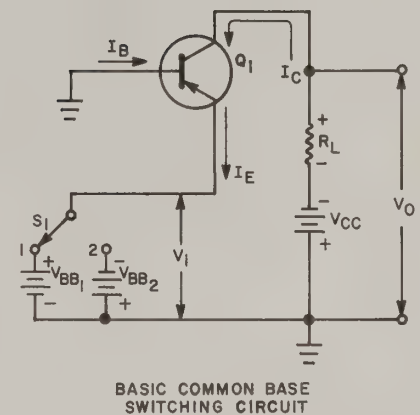


Figure
21

impedance is the lowest of the three circuits and its output impedance is the highest.

In terms of amplification, the common emitter stage has the most power gain because both voltage and current gain are much greater than unity. In general, the common base circuit has a little higher voltage gain than the common emitter, but the current gain is a little less than one. Therefore, the power gain is much less than that of the common emitter circuit.

Like the common collector switching circuit, the common base switching circuit does not produce any polarity reversal between the input and output voltage waveforms. In Figure 21, when S_1 is in position 1 the input voltage V_i is positive with respect to ground. Under these conditions, Q_1 conducts and V_o is zero. Conversely, when S_1 is in position 2, V_i is negative with respect to ground. Under these conditions, Q_1 is cut off and V_o is equal to $-V_{CC}$. Since the collector of Q_1 is negative with respect to ground, V_o is negative. Therefore, when V_i is positive V_o is zero, and when V_i is negative V_o is negative.

Temperature Effects

The operating temperature of a transistor switching circuit has a definite effect on circuit behavior. When a transistor is operated as a switch, it operates in one of three regions; active, cutoff, or saturation. As explained earlier, when a transistor is operated in its saturated region, the power dissipated is very low. For this reason, temperature has little effect on the power dissipation of a saturated transistor. However, when a transistor is operated in its cutoff or active region, temperature has a much greater effect on transistor power dissipation and circuit operation.

THERMAL RUNAWAY can occur when a transistor is operated in its active region. In its active region, a transistor has its emitter junction forward biased and its collector junction reverse biased. Collector current depends upon base current. As base current increases, collector current increases. Collector current causes the transistor to heat up. At low operating temperatures, the transistor can dissipate the heat of a normally operating transistor into the surrounding air. However, if the temperature of the air surrounding the transistor is high, the transistor is less able to dissipate the heat it is producing. Thus, the temperature of the transistor increases.

An increase in transistor temperature causes the collector current to increase. As a result, still more power is developed in the collector circuit and the temperature increases further. This condition can continue until the transistor is destroyed by excessive heat.

Figure 22 shows a PNP transistor switch, common emitter configuration, operated in its active region. The emitter junction is forward biased by V_{BB} , and the collector junction is reverse biased by V_{CC} . R_E in the emitter circuit prevents thermal runaway. The voltage across R_E developed by I_C makes the emitter of Q_1 negative with respect to ground. Notice that the voltage applied between the base and emitter of Q_1 not only is determined by V_{BB} , but also by the voltage across R_E . The voltage across R_E opposes the voltage of V_{BB} and tends to lower the forward bias on the emitter junction of Q_1 .

If I_C increases due to an increase in temperature, the voltage across R_E increases. Since V_{R_E} opposes V_{BB} , the forward bias on Q_1 is reduced. As a result, the amount of I_C increase is reduced. Thus, R_E tends to prevent any increases in collector current and thermal runaway is prevented. The value of R_E is kept low so, at normal operating temperatures, its voltage drop does not greatly reduce the forward bias on the emitter junction of Q_1 .

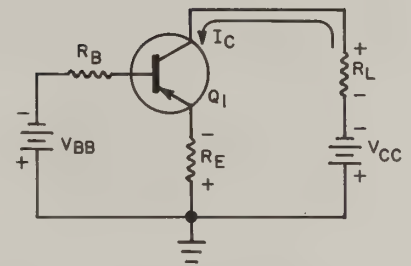
Temperature also affects the operation of common base and common collector transistor switching circuits. As with the common emitter configuration, the resistance in the base circuit of a transistor in the common base or common collector configuration should be low to prevent large collector currents and thermal runaway when the transistor is cut off. The load resistor is in the emitter circuit in the common collector configuration. Thus, the circuit itself provides good thermal runaway protection. The resistor placed in the emitter lead of Figure 22, to prevent thermal runaway in the active region, may also be used in the common base configuration.

Selection of Transistor Types

Throughout the discussion of transistor switching, the importance of switching time has been stressed. This and other characteristics depend on the type of transistor used. There are a number of ways of forming transistor junctions. Each type of transistor has its advantages and disadvantages as far as switching time, power-handling ability, and voltage requirements are concerned.

The GROWN JUNCTION TRANSISTOR was one of the first types to be developed. The switching time of a junction type transistor depends largely on the thickness of the base region. With the growing process, very thin base regions are difficult to obtain. As a result, grown junction transistors are used only in switching circuits which are not required to have very short switching times.

A modified process may be used to produce a GROWN DIFEUSED JUNCTION TRANSISTOR with a very thin base region, and thus a short



USE OF EMITTER RESISTOR
TO DECREASE TEMPERATURE
SENSITIVITY

Figure
22

SWITCHING CHARACTERISTICS

switching time. An alternate method is sometimes used to produce a DIFFUSED MELTBACK JUNCTION TRANSISTOR. With this process, very thin base regions can also be produced.

Grown junction transistors have been produced with cutoff frequencies approaching 10 MHz. The power-handling abilities of these transistors are usually below 1 watt. The diffused and meltback type transistors have cutoff frequencies approaching 100 MHz. They are very suitable for use in high speed switching circuits. As with the simple grown junction transistors, diffused and meltback type transistors usually have power-handling capacities below 1 watt.

An entirely different technique, resulting in an ALLOY JUNCTION TRANSISTOR, has a very large base region. Such transistors have power-handling capabilities of around 200 mW, and a frequency response of around 12 MHz.

Another type of transistor which is especially suited for use in high speed switching circuits is the SURFACE BARRIER TRANSISTOR. This type of transistor has a short switching time. Surface barrier transistors have been produced with cutoff frequencies from about 50 to 100 MHz. However, due to the very narrow base region, the power-handling ability of a surface barrier transistor is usually lower than 100 milliwatts. A modification of the surface barrier transistor called the MICROALLOY DIFFUSED BASE TRANSISTOR (MADT) has a cutoff frequency near 270 MHz.

A DIFFUSED JUNCTION TRANSISTOR also provides extremely short switching times, but has little power-handling capability. Power handling may be increased by placing the base region on a raised portion; above the collector region. This results in a MESA TRANSISTOR.

Field effect transistors may also be used as switches. Either N-channel or P-channel devices may be used, but result in switching times considerably greater than those obtained with high-speed junction transistors. Junction FET's, in general, exhibit longer switching times than metal-oxide-semiconductor (MOSFET) types. Circuit arrangements similar to those described for junction transistors are used with the FET source replacing the junction transistor emitter; the FET gate replacing the base; and the FET drain replacing the collector.

VACUUM TUBE CONSIDERATIONS

Multi-element vacuum tubes can also be used as switches. A triode tube switching circuit that is very similar to the common emitter switching cir-

cuit of Figure 15 is shown in Figure 23. In Figure 23, when a negative input voltage is applied, the grid of V_1 is driven more negative and V_1 cuts off. With V_1 cut off, plate current I_b is zero. As a result, the voltage across R_L is zero and the plate voltage of V_1 is equal to E_{bb} . Since the output is taken from the plate of V_1 to ground, V_o is equal to E_{bb} and the circuit is in its 1 state.

When a positive input signal is applied, the grid of V_1 is less negative and V_1 conducts. Plate current through R_L develops a voltage across R_L which subtracts from E_{bb} and lowers the plate voltage of V_1 . The circuit is now in its 0 state, since there is a minimum output voltage. The circuit of Figure 23 acts similar to the circuit of Figure 15. In both circuits V_o is minimum when the device conducts and V_o is maximum when the device is cut off. Thus, the circuits of Figures 15 and 23 both act as shunt type switching circuits.

The vacuum tube switching circuit of Figure 23 does not possess any storage properties, therefore does not have any storage time. However, besides the shunt circuit capacitance affecting its switching time, ELECTRON TRANSIT TIME affects the switching time of the circuit. This is the time required for electrons to travel from the cathode to the plate. Electron transit time limits the speed at which the circuit can change states. Therefore, it sets a maximum limit on the switching speed of a vacuum tube switching circuit.

Vacuum tube diodes may be used as switches in the place of semiconductor diodes. Operation is very similar. Only a few characteristic differences will be observed. The characteristic curve for a vacuum tube diode is similar to that of a semiconductor diode, except that the breakdown point of a vacuum tube diode is usually much higher than that of a semiconductor diode. Some semiconductor diodes break down under a few volts of reverse bias, whereas several hundred or more volts of reverse bias may be required to break down a vacuum tube diode. For this reason, reverse switching is seldom used with vacuum tube diodes.

As with carrier storage in semiconductor diodes, transit time in vacuum tube diodes is only troublesome in very high speed switching circuits. Special tubes with closely spaced electrodes are used in some circuits to reduce transit time and, thus, permit higher speed switching.

SUMMARY

The diode used as a switch has two operating states. It is either "ON" or "OFF". In terms of logical (binary) digits, these states are 1 and 0—with the assignment completely arbitrary. Switching circuits are of either the series or shunt type. For the series type, the maximum output voltage occurs

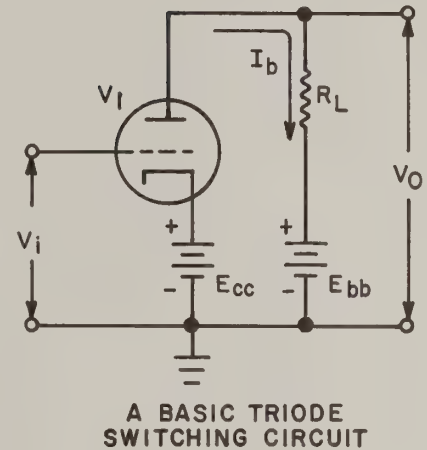


Figure
23

when the diode is conducting; the minimum output voltage occurs when the diode is cut off. For the shunt type, the reverse is true.

In either the diode series or shunt-type circuit one of two switching methods (forward or reverse switching) may be used. For forward switching, the diode is alternately biased from the forward region to the reverse cutoff region and back again. However, the reverse polarity amplitude must not exceed the diode breakdown voltage. For reverse switching, a zener diode is used only in the reverse region, alternating between breakdown and cutoff. This requires a single polarity signal with sufficient amplitude to produce breakdown.

Diode switching is not ideal. That is, the conducting characteristic offers a definite small resistance to current, while the cutoff characteristic does not provide infinite resistance. Furthermore, shunt capacitance in the circuit does not allow an instantaneous switching action. Changes in voltage require a short time to build up or decay. In addition, carrier storage effects introduce a recovery time delay anytime the diode is switched from the forward region into the reverse region.

Transistor switching circuits can be connected in three circuit configurations: common base, common collector, and common emitter. The common emitter configuration is most widely used because of its high power amplification properties. Each transistor configuration has an input impedance which is different from its output impedance. Therefore they can be used as impedance matching devices. Besides performing the function of a switch, the transistor provides amplification.

As in diode switching circuits, circuit capacitance prevents a transistor switching circuit from changing states instantaneously. Carrier storage also occurs in transistor switching circuits. Just as in a diode switching circuit, carrier storage increases switching time.

The operating temperature of a transistor switching circuit has an effect on circuit operation. Unless precautions are taken, increases in temperature can cause thermal runaway resulting in the destruction of the transistor.

Vacuum tube diodes and triodes, as well as other multi-element tubes, can be used as switches. Switching circuit arrangements and operation are very similar. Although there are no carrier storage effects in vacuum devices, the electron transit time characteristic produces similar switching delays. The breakdown point for a vacuum tube diode is generally several hundred volts or more. Hence, reverse switching is seldom used with vacuum tube diodes.

The following Practice Exercise questions cover the subjects which you have just studied. They are:

TRANSISTOR SWITCHING

COMMON BASE SWITCHING

TEMPERATURE EFFECTS

SELECTION OF TRANSISTOR TYPES

VACUUM TUBE CONSIDERATIONS

32. A common base circuit can go into both voltage and current saturation. True or False?
33. In this discussion of Figure 21, the logic was (a) positive, (b) negative.
34. How do the input and output impedances of the common base stage compare to that of the common emitter configuration?
35. Why is it possible for a slight increase in temperature to result in the destruction of a transistor?
36. How does R_E in Figure 22 help to prevent thermal runaway?
37. Diffused and meltback type transistors are more suitable than grown junction types for high speed switching. True or False?

38. Alloy junction transistors can handle high power, but have a very _____ switching time characteristic.
39. How do JFET and MOSFET switching times compare with respect to one another and with respect to high-speed junction transistors?

IMPORTANT DEFINITIONS

ELECTRON TRANSIT TIME—The time required for electrons in a vacuum tube to travel from the cathode to the anode.

FALL TIME—The portion of a junction diode **RECOVERY TIME** where the diode reverse current decays from a large constant value to the saturated reverse current level. For transistor switching circuits, it is considered to be the time between the 90% and 10% points as the collector current decays to minimum.

FORWARD SWITCHING—A method for using the diode as a switch, requiring that the voltage across the diode alternate in polarity, driving the diode back and forth between forward conduction and reverse cutoff.

RECOVERY TIME—The time required for a diode current to stabilize at the saturated reverse current level when it is switched from forward conduction to reverse cutoff.

REVERSE SWITCHING—A method for using the diode as a switch, requiring a single polarity voltage with sufficient amplitude variation to drive the diode back and forth between reverse breakdown and reverse cutoff.

RISE TIME—The time required for the device output current to rise from 10% to 90% of its maximum value.

STORAGE TIME—A portion of the junction diode **RECOVERY TIME** where the diode reverse current is a large constant value due to carrier storage effects.

THERMAL RUNAWAY—A condition where an increase in transistor temperature causes an increase in collector current, which, in turn, causes a further increase in transistor temperature. Unless steps are taken to prevent thermal runaway, it can continue until the transistor is destroyed.

TURN-OFF DELAY—See **TURN-OFF TIME**.

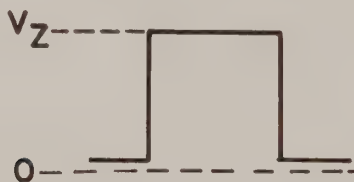
TURN-ON DELAY—See **TURN-ON TIME**.

TURN-OFF TIME—The total delay due to storage time and fall time.

TURN-ON TIME—The total delay due to switching delay time and rise time.

PRACTICE EXERCISE SOLUTIONS

1. series
2. two
3. False—When a diode is conducting, it presents some definite low resistance, and when it is cut off, it has some definite high resistance. Thus, it does not act as a perfect switch.
4. (a) conducting.
5. (b) cut off.
6. False—The internal resistance of a diode is very low when operated in its forward region, and thus it acts like a closed switch.
7. True
8. True
9. (b) negative.
10. False
11. True
12. (a) forward biased.
13. “Sweeping out” the majority carriers is the cause of the large pulse of reverse current through a semiconductor diode when it is switched from forward to reverse bias.
14. True
15. False—For either forward or reverse switching, the series-type output is in the 0 state when the diode is cut off; in the 1 state when the diode is conducting. This can be seen by comparing Figures 2, 12 and 13.
16. For a diode reverse switching circuit, the applied voltage must be of a single polarity and its magnitude must be small enough to prevent current flow from exceeding the maximum rated zener current.
- 17.



PRACTICE EXERCISE SOLUTIONS (Continued)

18. True
19. A transistor switching circuit may be connected in the common emitter configuration, the common base configuration, or the common collector (emitter follower) configuration.
20. low, closed
21. high, open
22. False—When operated as a saturated switch, a transistor operates in the cutoff and saturated regions. In saturation the collector voltage is quite low, resulting in low power dissipation. In the cutoff region, collector current is very low, resulting in low power dissipation. Thus, power dissipation is low in both of these operating states. On the other hand, an unsaturated transistor switching circuit operates in the active and cutoff regions. Although power dissipation is low in the cutoff region, it is quite high in the active region. As a result, an unsaturated switching circuit dissipates more power than a saturated switching circuit.
23. (b) shunt-type switch.
24. (b) current carrier storage.
25. Circuit capacitances predominate in the determination of turn-on time.
26. overdriving
27. input signal
28. True
29. load resistor, series
30. False
31. high, low
32. True
33. (b) negative.
34. Compared to the common emitter circuit, the input impedance of the common base is much lower and its output impedance is higher.
35. A slight increase in temperature, under the proper conditions, may cause the transistor base and collector currents to increase enough to increase

PRACTICE EXERCISE SOLUTIONS (Continued)

the transistor junction temperature. This causes further increases in the currents, more power dissipation and an even higher junction temperature. If not controlled, the transistor can be destroyed.

- 36. The voltage across R_E opposes the forward bias due to thermal current increases through R_E . Hence, it becomes possible to prevent the runaway condition.
- 37. True
- 38. poor
- 39. JFET switching times are longer than MOSFET switching times; however, neither FET can achieve the extremely short intervals associated with high-speed junction transistors.

1. B - A VACUUM TUBE OR SEMICONDUCTOR ACTS AS A SWITCH BECAUSE ITS INTERNAL RESISTANCE IS -- LOW DURING CONDUCTION AND HIGH DURING CUTOFF.

The internal resistance of a mechanical switch is either 0 (very low) or infinite (very high). The vacuum tube and semiconductor display similar characteristics; therefore, they can be used as switches. However, since vacuum tubes and semiconductors are not perfect conductors, they will always have some measurable amount of resistance when conducting, and they will never achieve infinite resistance when cut off.

2. B - THE TIME REQUIRED FOR A SWITCHING CIRCUIT TO CHANGE FROM ONE OPERATING STATE TO ANOTHER IS CALLED -- SWITCHING TIME.

It takes an amount of time for a switching circuit to switch from one state to the other. The time required is called the switching time of the circuit.

3. B - RECOVERY TIME IS THE TOTAL TIME INTERVAL OF THE REVERSE CURRENT PULSE DEVELOPED BY A DIODE WHEN IT IS SWITCHED FROM -- FORWARD TO REVERSE BIAS.

When a diode is forward biased, current carriers are introduced into both the P and N-type materials. When the diode is reversed biased, these carriers are temporarily "trapped" in a condition where only minority carriers can flow. Before equilibrium is reached, there is a momentary high reverse current through the diode. The time interval required to establish equilibrium is known as the Recovery Time of the diode.

4. D - THE LARGE PULSE OF REVERSE CURRENT PASSED BY A DIODE AS IT IS SWITCHED FROM FORWARD TO REVERSE BIAS IS A RESULT OF -- CURRENT CARRIERS.

When a diode is forward biased, current carriers are introduced into both the P and N-type materials. When the diode is reversed biased, these carriers are temporarily "trapped" in a condition where only minority carriers can flow.

5. C - THE TRANSISTOR CIRCUIT CONFIGURATION THAT HAS THE HIGHEST POWER GAIN IS THE -- COMMON EMITTER.

In a common emitter amplifier, the input signal causes variations of base current. In turn, the base current causes collector current to vary. Therefore, the common emitter circuit configuration provides the highest power gain, and thus delivers a large output signal.

6. B - WHEN Q_1 IN THE COMMON EMITTER SWITCHING CIRCUIT OF FIGURE 15 IS OPERATED IN ITS CUTOFF AND ACTIVE REGIONS, IT IS OPERATING AS -- AN UNSATURATED SWITCHING CIRCUIT.

To operate as a saturated switch, Q_1 would have to be operated in the saturation region. When Q_1 is operated in the cutoff and active regions it can only operate as an unsaturated switch.

7. C - THE TURN-ON TIME OF A TRANSISTOR SWITCHING CIRCUIT IS THE TIME REQUIRED FOR COLLECTOR CURRENT TO -- INCREASE FROM 0 TO 90% OF ITS MAXIMUM VALUE.

In transistor switching circuits, turn-on time is usually defined in terms of collector current. It is the time required for collector current to increase from its cutoff value to 0.9% of its maximum value.

8. B - CURRENT CARRIERS ARE STORED (INTRODUCED) IN THE BASE REGION OF A TRANSISTOR WHEN -- IT IS CONDUCTING.

When a transistor is forward biased, thus conducting, current carriers are being introduced or stored in the base region.

9. B - IF I_C IN FIGURE 22 INCREASES TOO FAR AS A RESULT OF A TEMPERATURE INCREASE, THE VOLTAGE ACROSS R_E INCREASES AND -- OPPOSES THE EMITTER JUNCTION FORWARD BIAS. Since V_{R_E} opposes V_{BB} , the forward bias on Q_1 is reduced.

10. B - VACUUM TUBE SWITCHING CIRCUITS -- ARE DELAYED BY ELECTRON TRANSIT TIME, JUST AS SEMICONDUCTOR SWITCHING CIRCUITS ARE DELAYED BY CARRIER STORAGE TIME.

Electron transit time is the time required for electrons to travel from the cathode to the plate. Electron transit time limits the speed at which the circuit can change states. Therefore, it sets a maximum limit on the switching speed of a vacuum tube switching circuit.

QUESTIONS

IMPORTANT—These instructions **MUST** be accurately followed to avoid loss, or errors in grading.

Indicate your answer on this sheet by filling in the box for the most correct answer to each question, as illustrated in the example below.

When all questions have been answered, place the answer card in the proper position to line up the boxes on the card with the boxes on the sheet.

Next, copy the complete lesson code into the space provided on the card, and fill in the answer boxes to correspond with those previously filled in on this sheet.

Before mailing, be certain your correct student number, name and address appear on the card.

LESSON CODE
5210A

Example: A five-sided figure is a

A	☐
B	☐
C	☒
D	☐

(A) hexagon. (B) triangle. (C) pentagon. (D) octagon.

1.

A	☐
B	☒
C	☐
D	☐

 A vacuum tube or semiconductor acts as a switch because its internal resistance is (A) high during conduction and cutoff. (B) low during conduction and high during cutoff. (C) low during conduction and cutoff. (D) the same whether conducting or cut off.

2.

A	☐
B	☒
C	☐
D	☐

 The time required for a switching circuit to change from one operating state to another is called (A) reverse phase. (B) switching time. (C) delay time. (D) recovery phase.

3.

A	☐
B	☒
C	☐
D	☐

 Recovery time is the total time interval of the reverse current pulse developed by a diode when it is switched from (A) zero to forward bias. (B) forward to reverse bias. (C) reverse to forward bias. (D) reverse to zero bias.

4.

A	☐
B	☐
C	☒
D	☐

 The large pulse of reverse current passed by a diode as it is switched from forward to reverse bias is a result of (A) the forward resistance of the diode. (B) lead inductance. (C) circuit capacitance. (D) current carrier.

5.

A	☐
B	☐
C	☒
D	☐

 The transistor circuit configuration that has the highest power gain is the (A) common cathode. (B) common collector. (C) common emitter. (D) common base.

6.

A	☐
B	☒
C	☐
D	☐

 When Q_1 in the common emitter switching circuit of Figure 15 is operated in its cutoff and active regions, it is operating as (A) a saturated switching circuit. (B) an unsaturated switching circuit. (C) a forward switch. (D) a series switching circuit.

7.

A	☐
B	☐
C	☒
D	☐

 The turn-on time of a transistor switching circuit is the time required for collector current to (A) increase from 0 to 50% of its maximum value. (B) decrease from maximum to 10% of its maximum value. (C) increase from 0 to 90% of its maximum value. (D) decrease from 100% to 50% of its maximum value.

8.

A	☐
B	☒
C	☐
D	☐

 Current carriers are STORED (introduced) in the base region of a transistor when (A) base and collector current are zero. (B) it is conducting. (C) it is cut off. (D) the emitter and collector junctions are reverse biased.

9.

A	☐
B	☒
C	☐
D	☐

 If I_C in Figure 22 increases too far as a result of a temperature increase, the voltage across R_E increases and (A) the emitter junction of Q_1 becomes reverse biased. (B) opposes the emitter junction forward bias. (C) the collector junction of Q_1 becomes reverse biased. (D) Q_1 can go into cutoff.

10.

A	☐
B	☒
C	☐
D	☐

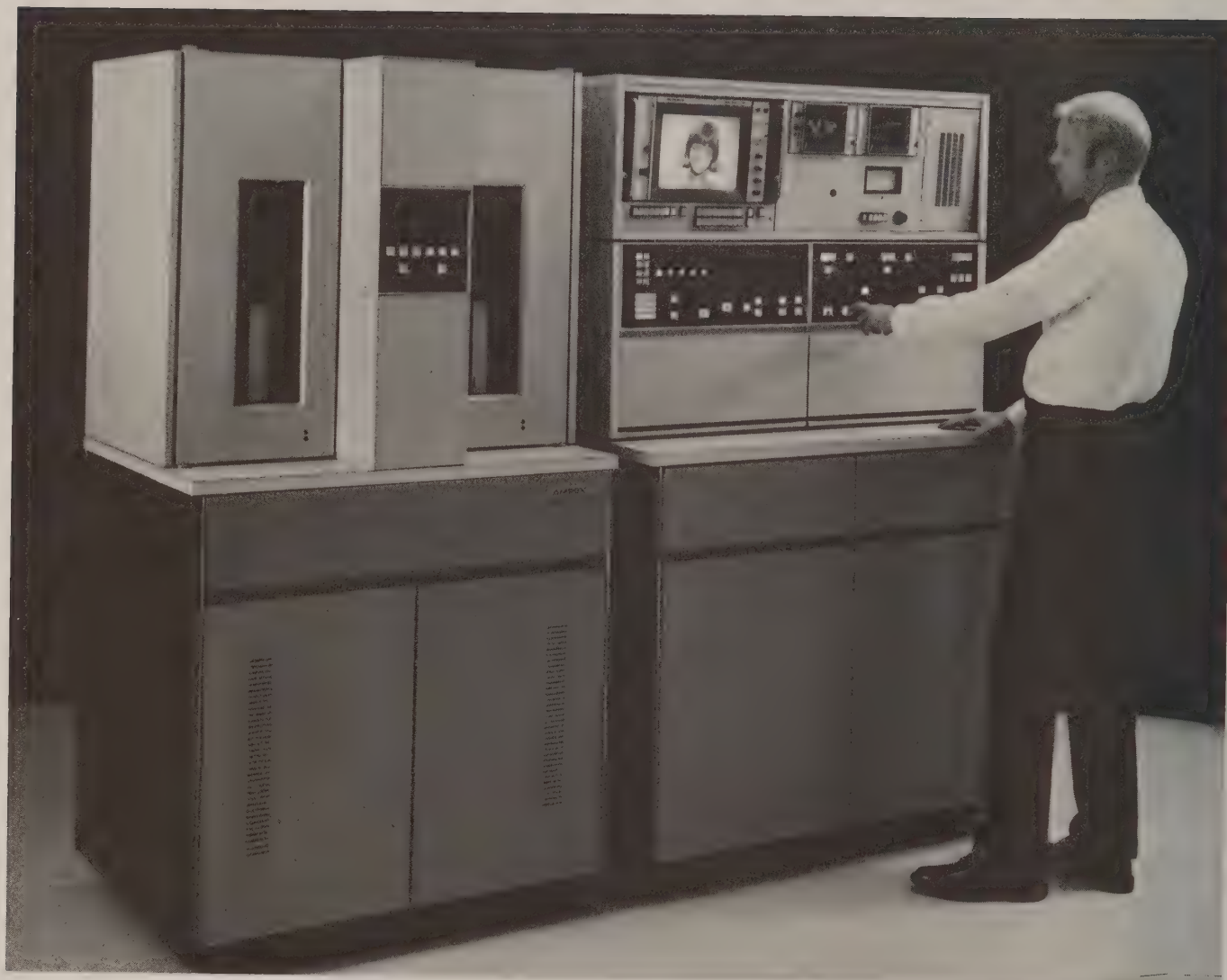
 Vacuum tube switching circuits (A) frequently take advantage of the reverse breakdown characteristic. (B) are delayed by electron transit time, just as semiconductor switching circuits are delayed by carrier storage time. (C) do not exhibit any switching delay characteristics. (D) can only be operated by negative signals, applied at the grid.

DIODE LOGIC CIRCUITS

5215

© COPYRIGHT 1973
BELL & HOWELL SCHOOLS, INC.
ALL RIGHTS RESERVED





Operation of the automatic video cassette recorder/reproducer shown here is made possible because of the many digital logic circuits it contains. These circuits are mounted on plug-in boards for ease of servicing. This equipment gives the broadcaster his continuing choice of random or sequential programming of up to 40 events from as many as 24 cassettes. It provides excellent color fidelity and permits last-minute push-button changes of programming.

*Courtesy Ampex Corp.
Audio-Video Systems Division*

CONTENTS

The AND and OR Functions	Page 5
Diode AND Gates	Page 7
Diode OR Gates	Page 12
Circuit Losses	Page 15
Diode-Transistor Logic	Page 18
Use of Noninverting Amplifiers	Page 18
Use of Inverting Amplifiers	Page 21
Low Level Logic	Page 24
Tunnel Diode Switching	Page 26

*We know what we are, but know not what
we may be.*

—William Shakespeare

DIODE LOGIC CIRCUITS

Most digital systems, like many other types of modern electronic systems, depend upon complex switching networks for their operation. The modern telephone switching system is an example of a complex switching network. In fact, problems encountered in the design of automatic telephone systems led to the first systematic studies of switching logic. Early automatic switching systems were based entirely on the use of electromagnetic relays and stepping switches. However, these relays and switches operate far too slowly for use in most present-day digital switching systems. They have been replaced by tubes, semiconductor diodes, transistors, other solid state and magnetic (ferrite) devices. This lesson is devoted to the use of semiconductor diodes and transistors.

The main advantage of a diode logic circuit is its simplicity. However, there are a few disadvantages as well. For example, a diode does not provide amplification. As a result, a signal is attenuated and its level shifted as the signal passes through the diode logic circuit. By the time a signal passes through a number of diode logic circuits, it may be reduced to an unusable level. However, this disadvantage can be eliminated by inserting voltage amplifiers between the diode logic circuits to restore the signals to their proper levels.

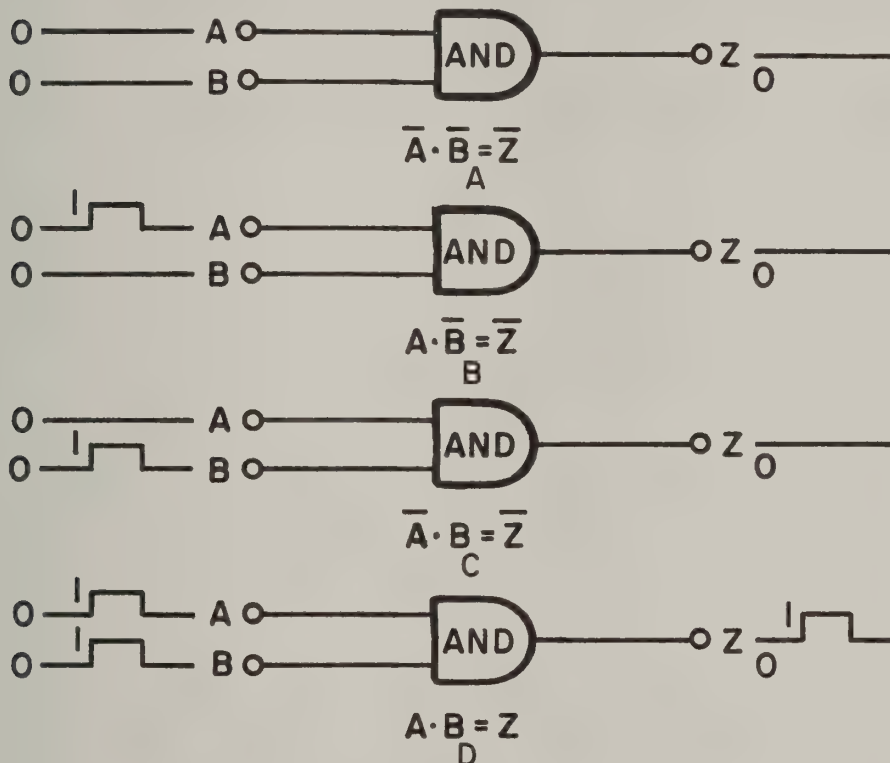
Voltage amplifiers usually follow a group of two or three diode logic circuits. Where signal levels are critical, or where losses are high, a voltage amplifier may be inserted after each diode logic circuit. The amplifiers increase the cost and complexity of the system, but the increased signal power more than compensates for the added cost and complexity.

Semiconductors are widely used because of their small size, low heat dissipation and high reliability over a long operating life. However, vacuum tubes are used in some digital systems. AND and OR gates can be built around diodes. A NOT circuit requires an amplifier that produces voltage signal inversion, and therefore it cannot be built with conventional diodes alone.

The information processed by a digital system is usually in the form of a binary code. In turn, the binary-coded information is represented by voltage levels. For example, in a digital system the binary digit 0 (0 state) may be represented by a -5 volt dc level, and the binary digit 1 (1 state) by a +5 volt dc level. When dc levels such as these are used to represent binary digits, the arrangement is called dc or **STATIC LOGIC**. Another digital system may use a positive pulse to represent a binary digit 1 and a negative pulse to represent a binary digit 0. When pulses are used to represent binary digits, the arrangement is called ac or **DYNAMIC LOGIC**.

THE AND AND OR FUNCTIONS

Two basic relationships exist when a switching circuit contains more than one switching element. These two relationships are referred to as "AND" and "OR". The AND relationship is also referred to as a conjunction, an intersection, or logical multiplication. The OR relationship is also referred to as a disjunction, a union, or logical addition.



FUNCTIONAL OPERATION OF
AND CIRCUIT

Figure 1

A "functional" method for describing the AND operation is presented in Figure 1. This was described in an earlier lesson. There is an output signal (1 state) only if there is an input signal (1 state) at terminal A and terminal B at the same time. No other input combination will produce a 1 state output signal in this type of circuit.

$$Z = A \cdot B \quad (\text{AND}) \quad (1)$$

Since A and B each has two possible values (0 and 1), these values may be combined in 2^2 or 4 different ways. A binary logic circuit has 2^n possible input combinations where n is the number of inputs. If the AND gate had three inputs, there would be 2^3 or 8 possible combinations. If it had four inputs, there would be 2^4 or 16 possible combinations. Table 1 shows these

INPUT POINTS		OUTPUT POINT
A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1

Truth Table for AND
Function

Table
1

four combinations of A and B and the corresponding values of Z. Input and output points are not normally labeled on truth tables. This is the AND truth table, which shows that A and B both must be 1 (the input logic level 1 voltage in Figure 1 must be present in both) in order for Z to be 1. All other combinations of A and B cause Z to be 0.

$\bar{A}$ and $\bar{B}$ are read as NOT A and NOT B, respectively. Since only two states are possible, 1—NOT 1—must be 0 and 0—NOT 0—must be 1.

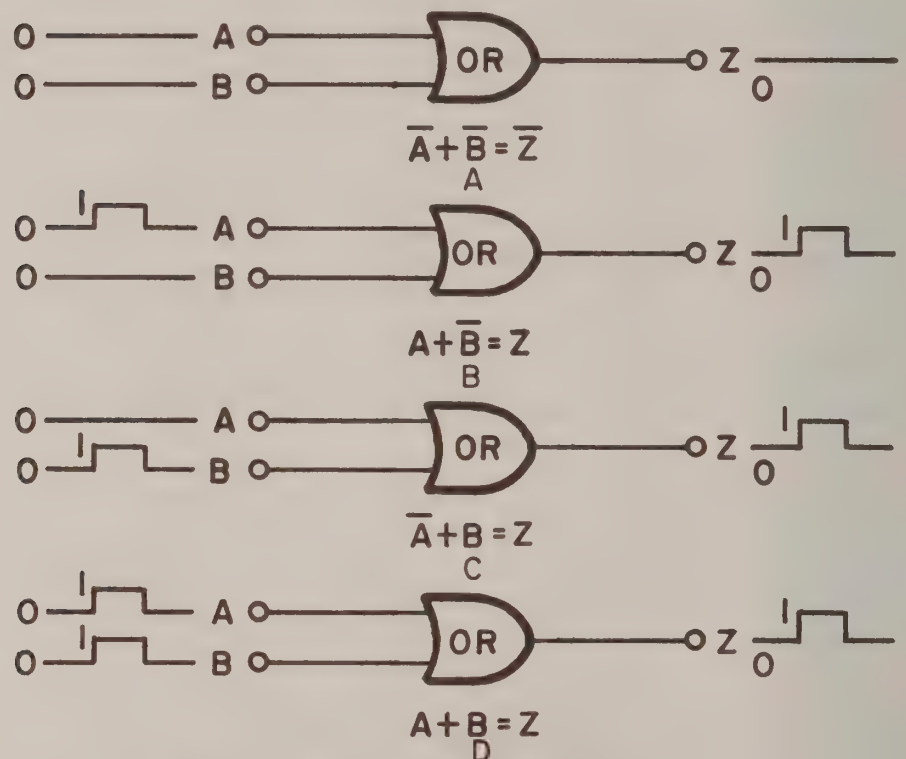


Figure 2

As mentioned in a previous lesson, the “functional” block method for describing the OR operation is presented in Figure 2. Note that the symbol for an OR gate is used in this figure. There is an output signal (1 state) when there is an input signal (1 state) at terminal A or at terminal B or at both terminals, simultaneously.

INPUT POINTS		OUTPUT POINT
A	B	Z
0	0	0
0	1	1
1	0	1
1	1	1

Truth Table for OR Function

$$Z = A + B \quad (\text{OR}) \quad (2)$$

Table 2

As with the AND operation previously described, since there are two inputs, each having two possible values, the values of A and B for Figure 2

may be combined in 2^2 or 4 different ways. Table 2 shows these combinations and the corresponding values of Z. This OR truth table shows that any appearance of a 1 at input A or input B, or any input logic level 1 voltage in Figure 2 will cause Z to be 1. Z will be 0 only when A and B are both 0; logic level 0 voltages are applied in Figure 2.

AND and OR circuits are not restricted to two inputs. Figure 3 shows a standard logic diagram for a three-input AND function and a four-input OR function. Many inputs may be assigned to each symbol. In expression form, the switching function is obtained by "dotting" the number of inputs for AND; by "summing" the number of inputs for OR. For example, the three-input AND function is $Z = A \cdot B \cdot C$ and the four-input OR function is $Z = A + B + C + D$.

Diode AND Gates

As mentioned in an earlier lesson, switching circuits that pass one or more signals from their input terminals to their output terminals only under certain conditions are often called gates. The gate is "open" when it passes a signal and "closed" when it does not pass a signal. Switching circuits are also called LOGIC CIRCUITS because they perform logical decisions. For example, the logical decision performed by an AND circuit is to indicate (by a 1 at the output) that all of its inputs are 1. The logical decision performed by an OR circuit is to indicate (by a 1 at the output) that at least one of its inputs is a 1.

A DIODE LOGIC circuit consists of a bias supply, a load resistor, and one diode for each input. Figure 4 shows the schematic diagram of a two-input diode logic circuit. The circuit contains two semiconductor diodes, D_1 and D_2 ; a diode load resistor, R_1 ; and a bias supply, V_S . Input signals at A and B are applied to the cathodes of D_1 and D_2 , respectively. The output at Z is the $D_1 D_2$ anode-to-ground voltage. Notice no ground connections are shown. This is the standard procedure used in drawing logic circuits. All signals are considered to be taken with respect to ground or a common point unless otherwise indicated.

To analyze circuit operation, assume that -5 volts is applied to both inputs A and B. The -5 volts applied at both input terminals makes the cathodes of both D_1 and D_2 negative. The anodes of D_1 and D_2 are connected to the positive terminal of the $+30$ volt bias source, V_S , through R_1 . With their anodes positive and cathodes negative, D_1 and D_2 are forward biased. Assuming the forward resistances of D_1 and D_2 are zero (characteristics of a perfect diode), the -5 volt input voltage levels are applied through the zero internal resistances of D_1 and D_2 , to output Z.

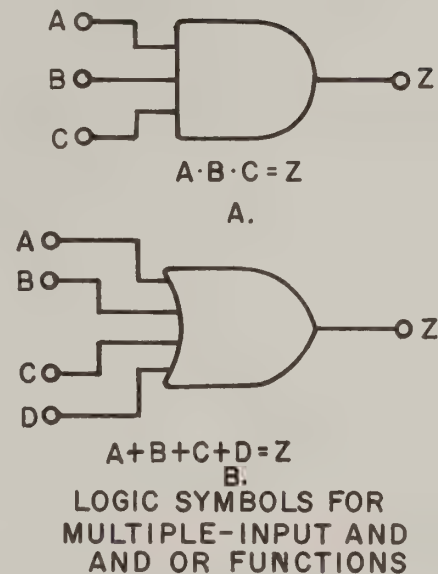


Figure 3

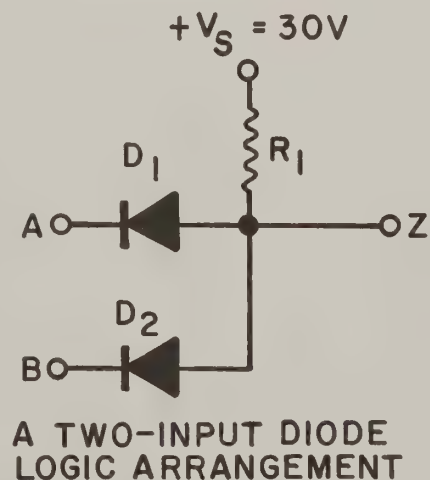
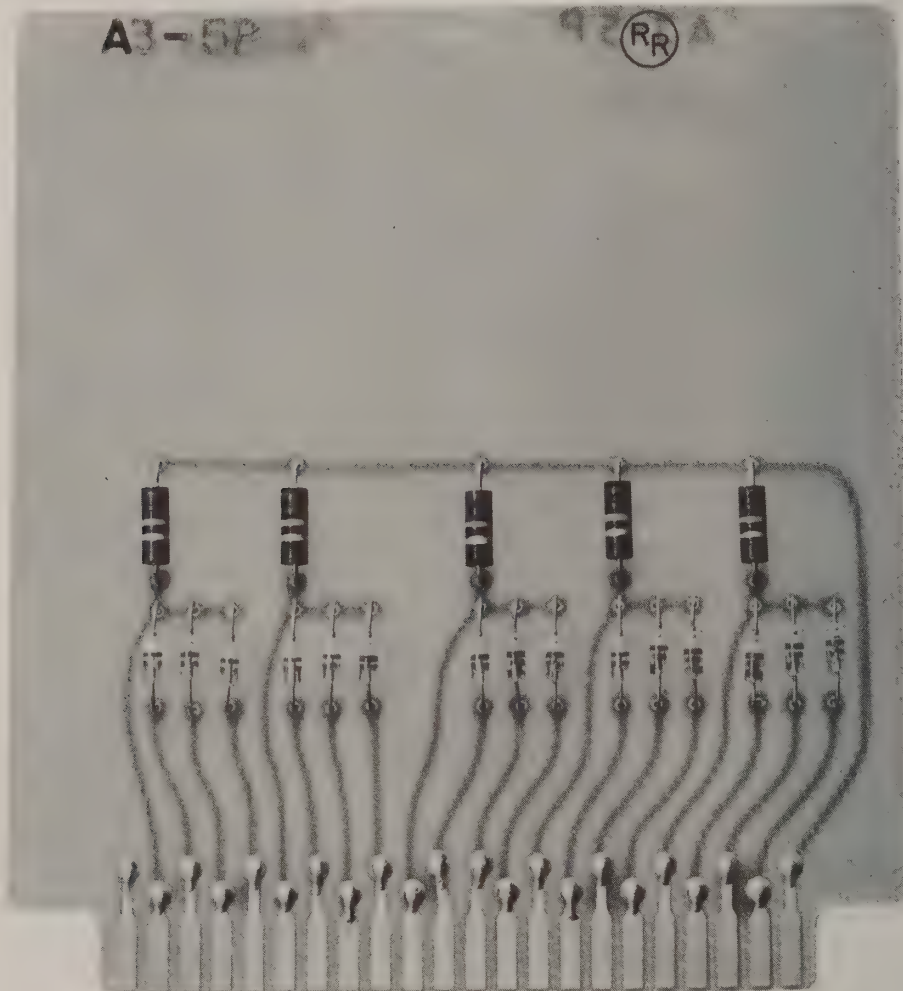


Figure 4

Now assume that input A is at -5 volts and input B is at $+5$ volts. The -5 volts at input A together with the $+30$ volt bias source forward biases D_1 . Again assuming that the forward resistance of D_1 is zero (characteristics of a perfect diode), the -5 volts at input A is applied to output Z. Notice that the anode of D_2 is also connected to the output terminal. The -5 volt output signal places the anode of D_2 at -5 volts with respect to ground. In addition, the $+5$ volts at input B makes the cathode of D_2 positive. As a result, D_2 is reverse biased. With D_2 reverse biased, its internal resistance is infinite (characteristics of a perfect diode). With infinite resistance, D_2 is effectively out of the circuit. The action is similar if input A is $+5$ volts and input B is -5 volts. Under these conditions, D_1 is reverse biased, effectively out of the circuit, and D_2 is forward biased. Again assuming zero forward resistance, the -5 volts at input B is applied to output Z.



Plug-in circuit board containing five three-input diode logic circuits.

Courtesy Random Research, Inc.

Now assume that both inputs A and B are at +5 volts. The +5 volts at both input terminals make the cathodes of D_1 and D_2 positive. The anodes of D_1 and D_2 are connected to the +30 volt bias source V_S through R_1 . Since their anodes are more positive than their cathodes, D_1 and D_2 are both forward biased. Assuming zero forward resistance for D_1 and D_2 , the +5 volts inputs at A and B are applied through D_1 and D_2 to output Z. Summarizing circuit action, output Z is at +5 volts only when both inputs A and B are at +5 volts. Whenever either or both inputs A and B are at -5 volts, the output Z is at -5 volts.

The input signal levels and the resultant output signal levels for the circuit of Figure 4 are shown in truth table form in Figure 5A. The logical operation performed by the diode gate can be determined from the truth table by choosing the 0 and 1 signal levels and substituting 0's and 1's into the truth table for the respective signals. Assume that the 0 voltage level is -5 volts and the 1 voltage level is +5 volts. All the +5 volt signals in the table of Figure 5A can be replaced with 1's and all the -5 volt signals with 0's. Performing this conversion produces the table shown in Figure 5B. Under these signal conditions the diode gate of Figure 4 operates as an AND gate. Output Z is 1 (+5 volts) only when both inputs A and B are 1 (+5 volts), and output Z is 0 (-5 volts) when either or both inputs A and B are 0 (-5 volts).

A gate circuit is often called a positive type gate when the binary digit 1 is a positive voltage, and a negative type gate when the binary digit 1 is a negative voltage. Using this convention, the diode gate circuit of Figure 4 is a **POSITIVE AND GATE**. It operates as an AND gate when the binary digit 1 is a positive voltage.

In some digital systems, it may be desirable to have a diode gate circuit that is a **NEGATIVE AND GATE**. Such a gate can be formed by reversing the diodes and the bias supply V_S in the circuit of Figure 4. Figure 6 shows this arrangement.

To examine circuit operation of Figure 6, assume that both inputs A and B are at +5 volts. The +5 volts at both inputs makes the anodes of D_1 and D_2 positive. The cathodes of D_1 and D_2 are connected to the negative terminal of the -30 volt bias source V_S . Therefore, both D_1 and D_2 are forward biased and the +5 volt input logic level 0 voltages are applied through D_1 and D_2 to output Z.

Now assume input A is at +5 volts and input B is at -5 volts. The +5 volts at input A, together with the -30 volt bias source V_S , forward biases D_1 . With D_1 forward biased, its internal resistance is zero (assume perfect diode) and the +5 volts at input A is applied through D_1 to output Z. The +5 volts at output Z is also applied to the cathode of D_2 . The -5 volts at input B to-

A	B	Z
-5	-5	-5
+5	-5	-5
-5	+5	-5
+5	+5	+5

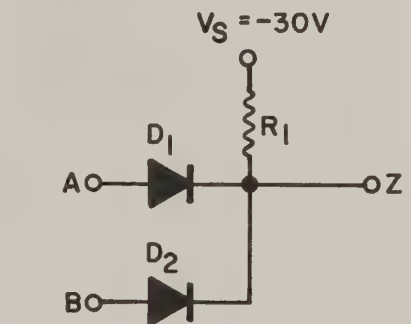
A

A	B	Z
0	0	0
1	0	0
0	1	0
1	1	1

B

TRUTH TABLE CONSTRUCTION FOR POSITIVE AND LOGIC ASSIGNMENT IN FIGURE 4 (0=-5 AND 1=+5)

Figure 5



LOGIC ARRANGEMENT OF FIGURE 4 WITH DIODES AND SUPPLY POLARITY REVERSED

Figure 6

A	B	Z
+5	+5	+5
-5	+5	+5
+5	-5	+5
-5	-5	-5

A

A	B	Z
0	0	0
1	0	0
0	1	0
1	1	1

B

TRUTH TABLE CONSTRUCTION
FOR NEGATIVE AND LOGIC ASSIGNMENT
IN FIGURE 6 (0=+5 AND 1=-5)

Figure
7

gether with the +5 volts at output Z reverse biases D_2 . The high internal resistance of reversed biased D_2 effectively disconnects it from the circuit. Thus, the +5 volts at input A causes D_1 to conduct and the -5 volts at input B cuts off D_2 . If the input signals are changed so that input A is at -5 volts and input B is at +5 volts, the output Z still remains at +5 volts. However, in this case D_1 is cut off and D_2 conducts.

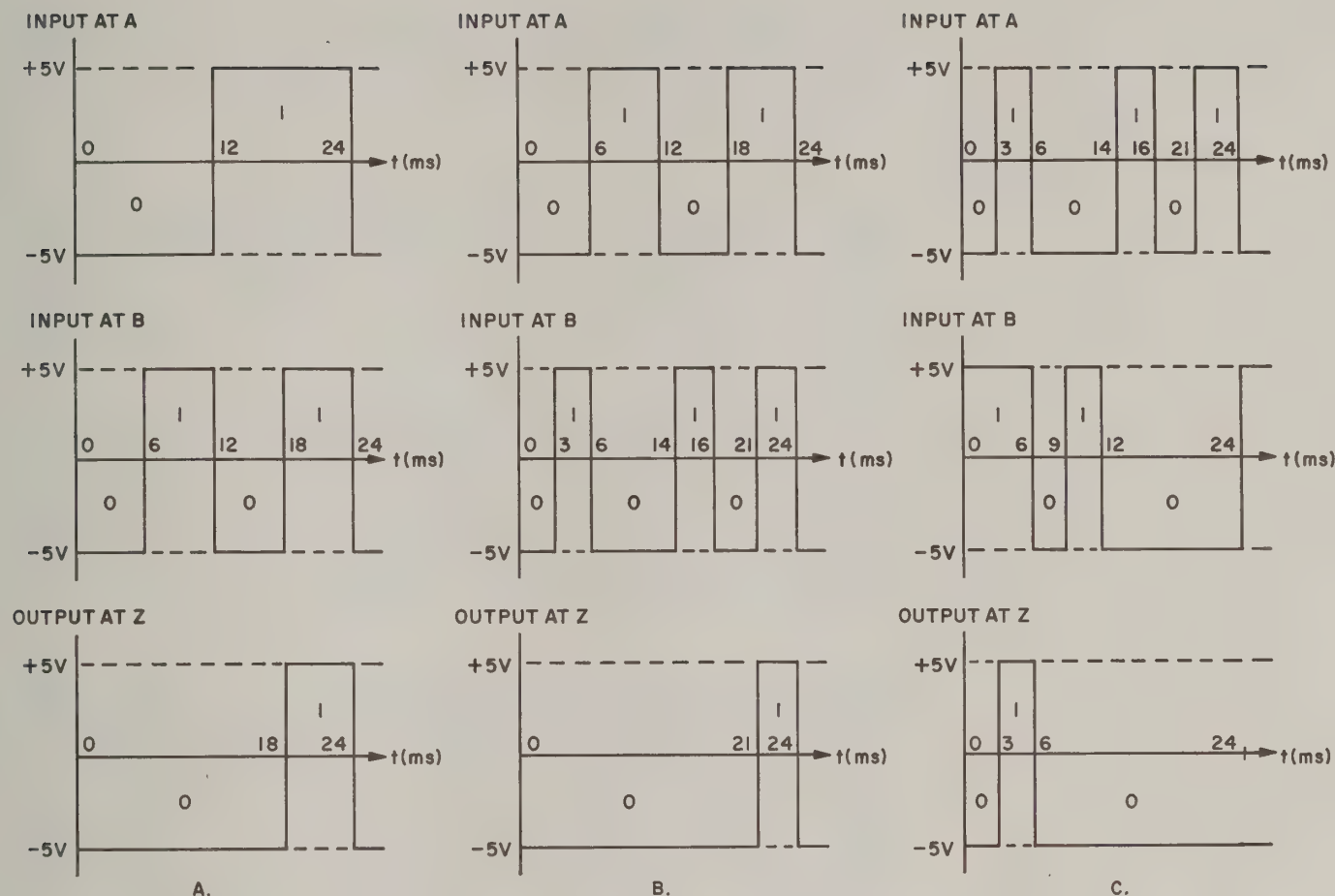
With -5 volts at both inputs A and B, the anodes of both D_1 and D_2 are negative. The cathodes of D_1 and D_2 are connected to the negative terminal of the -30 volt source. As a result, both D_1 and D_2 are forward biased; and the -5 volts at the input terminals is applied through D_1 and D_2 to output Z (still assuming perfect diodes).

The truth table of Figure 7A shows the input and resultant output signals for the diode gate circuit of Figure 6. The logical operations performed by this circuit can be examined in the same manner as the circuit of Figure 4. Choosing binary digit 0 equals +5 volts and binary digit 1 equals -5 volts yields the truth table of Figure 7B. Under these signal conditions, the diode gate circuit of Figure 6 operates as a negative AND gate. Output Z is 1 (-5 volts) only when both inputs A and B are 1 (-5 volts). When inputs A or B or both A and B are 0 (+5 volts), output Z is 0 (+5 volts).

Although dc levels were used for the binary digits 0 and 1 in the diode gates previously described, pulse signals also may be used. In this case, an AND gate delivers a binary digit 1 output pulse only when binary digit 1 pulses are simultaneously applied to all of its input terminals.

The waveforms in Figure 8 show the output responses, in relation to time, to several types of inputs of the positive AND gate. The circuit of Figure 4 will perform this operation since the output at Z is at +5 volts only when both inputs A and B are at 5 volts. However, when either or both inputs are at -5 volts, the output is also at -5 volts, as was previously described.

In Figure 8A, the input applied to A is a square wave signal with a period of 24 ms (24 milliseconds) and peaks at +5 and -5 volts. The input at B is twice the frequency (its period is half as long) with the same peak voltages. The two input signals are synchronized such that negative going edges of the input at A coincide with negative going edges of the signal at B. At $t = 0$, both edges are negative going. Thus, from $t = 0$ to $t = 6$ ms, both inputs are held at -5 volts. Consequently, output Z is also at -5 volts. However, from $t = 6$ to $t = 12$ ms, the input at B has changed to +5 volts. Input A is still at -5 volts; therefore, since both inputs are not at +5 volts, the level at Z remains at -5 volts.



POSITIVE AND RESPONSES FOR SEVERAL DIFFERENT INPUT TYPES

Figure 8

At $t = 12$ ms, both input signals make a polarity change with A becoming +5 volts and B becoming -5 volts. This time, the -5 volts at B causes the output Z to remain at -5 volts. Z remains at -5 volts until $t = 18$ ms when input B changes polarity.

From $t = 18$ to $t = 24$ ms, we have both inputs A and B at +5 volts. The output at Z consequently shifts to +5 volts and remains at this level as long as both inputs are at +5 volts. A look at the Figure 5A (bottom row) reveals this condition to be correct.

At $t = 24$ ms, both voltages at the inputs revert to -5 volts, causing the voltage at the output Z to revert back to -5 volts. This cycle is repeated every 24 ms.

In Figure 8B, the waveform that was at input B is now applied to input A. The signal at B consists of 3 ms wide pulses with spacing of 3, 8, and 5 milliseconds, respectively. The maximum input levels remain the same: +5 and -5 volts.

From $t = 0$ to $t = 3$ ms, both inputs are at -5 volts; thus, Z is at -5 volts. Between $t = 3$ and $t = 6$ ms, B is at +5 volts while A remains at -5 volts, thus causing Z to remain at -5 volts. From $t = 6$ to $t = 12$ ms, A is at +5 volts; however, B remains at -5 volts until $t = 14$ ms. Therefore, Z remains at -5 volts until at least $t = 14$ ms. From $t = 12$ to $t = 18$ ms, input A is at -5 volts; thus, Z is at -5 volts regardless of the voltage at B. From $t = 18$ to $t = 24$ ms, A is at +5 volts; however, it isn't until $t = 21$ ms that B becomes +5 volts. Therefore, from $t = 18$ to $t = 21$ ms, Z remains at -5 volts. From $t = 21$ to $t = 24$ ms, B is at +5 volts and A has been at +5 volts since $t = 18$ ms. Therefore Z is at +5 volts. After $t = 24$ ms, the cycle is repeated as in Figure 8A. This illustrates the basic operation of this AND circuit. When all inputs are at +5 volts, the output is at +5 volts; however, when any of the inputs are at -5 volts, the output is at -5 volts.

In Figure 8C, the input at B in Figure 8B was changed to A and the input B is now a 6 ms pulse followed by a 3 ms space and a 3 ms pulse. The period of repetition remains 24 ms in this example. From $t = 0$ to $t = 3$ ms, the output at Z is at -5 volts because the input at A is also at -5 volts. At $t = 3$ ms, the voltage at input A becomes +5 volts until $t = 6$ ms. Since the B input was at +5 volts from $t = 0$ to $t = 6$ ms, we find that from $t = 3$ to $t = 6$ ms, both inputs are at +5 volts. This causes output Z to be at +5 volts also for this interval. Nowhere else from $t = 6$ to $t = 24$ ms do we find both inputs at +5 volts; therefore, the output Z is at -5 volts for $t = 6$ to $t = 24$ ms.

The time scale in Figure 8 was in milliseconds. It could have been in seconds, microseconds, or in any other time scale.

The circuits we have described could have more than the two inputs shown. Each additional input requires an additional diode connected in the same direction as the other input diodes. All diodes are connected as those shown in Figures 4 and 6. If four diodes are connected in this way, the output is a binary digit (logical) 1 only when a logical 1 is applied to all four inputs. If the inputs are labeled A, B, C, and D, then the switching function for the circuit is $Z = A \cdot B \cdot C \cdot D$.

Diode OR Gates

Assume the logical 0 for the diode gate in Figure 4 is +5 volts and the logical 1 is -5 volts. To examine the logical operation under these signal conditions,

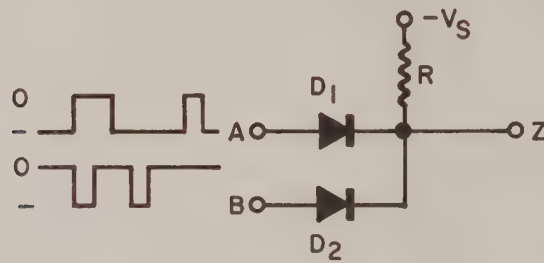
The following Practice Exercise questions cover the subjects which you have just studied. They are:

THE AND AND OR FUNCTION

DIODE AND GATES

1. What are the two possible basic relationships which exist when a switching circuit contains more than one switching element?
2. Write the switching function for "Z equals A AND NOT B".
3. If the three inputs to an AND gate were all 0, what would the expression for the switching function be?
4. Write the switching function for "Z equals A OR NOT B".
5. If a diode switching circuit, similar to Figure 4, has four inputs labeled A, B, C and D, all connected in parallel, what is the expression for its switching function?
6. In the circuit of Figure 4, if +5 volts is applied to input A and -5 volts is applied to input B, what is the voltage at output Z (assuming perfect diodes)?
7. In the circuit of Figure 4, if +5 volts is applied to both input A and input B, what is the voltage at output Z (assuming perfect diodes)?
8. Reversing the diode orientation and the supply polarity in a diode AND gate changes the logic _____.

9. How can additional inputs be added to the diode gate circuits of Figures 4 and 6?
10. Sketch the output you would expect from the circuit below with the input waveforms shown.



11. In the circuit of Figure 6, if -5 volts is applied to both inputs, what is the voltage at output Z (assuming perfect diodes)?

0's may be substituted for the +5 volt levels and 1's for the -5 volt levels in the truth table of Figure 5A. This results in the truth table of Figure 9A. Notice that output Z is now 1 (-5 volts) when either or both inputs A and B are 1 (-5 volts), and output Z is 0 (+5 volts) when inputs A and B are 0 (+5 volts). Therefore, under these signal conditions, the diode gate of Figure 4 operates as an OR gate. Also, since the circuit operates as an OR gate when the 1 signal is a negative voltage, the diode gate of Figure 4 is a **NEGATIVE OR GATE**.

The diode gate circuit of Figure 4 thus serves a dual purpose. When the 1 is a positive voltage and the 0 a negative voltage, the circuit operates as a positive AND gate. When the 1 signal is a negative voltage and the 0 signal is a positive voltage, the circuit operates as a negative OR gate.

A similar analysis may be applied to the diode gate of Figure 6. If the 0 and 1 signal levels are reversed, the diode gate circuit of Figure 6 operates as a **POSITIVE OR GATE**. Now the 0 is at -5 volts and the 1 is at +5 volts. Substituting 0's for the -5 volt signals and 1's for the +5 volt signals in the truth table of Figure 7A produces the truth table of Figure 9B. Notice that output Z is now 1 (+5 volts) when either or both inputs A and B are 1 (+5 volts), and output Z is 0 (-5 volts) when inputs A and B are both 0 (-5 volts).

The diode logic (DL) circuits of Figures 4 and 6 are very versatile. By changing the binary meanings of the voltage levels, the logical operation performed by the circuit is changed. Only two circuit types are required to form both positive and negative AND and OR gates. More inputs can be added to these circuits by adding more diodes through diodes in parallel with the existing inputs. However, as will be described later, the number of inputs is limited.

Figure 10 shows the output responses to several types of inputs in relation to time of the two-input positive OR gate. The circuit of Figure 6 can perform this operation since output Z is at +5 volts when either or both inputs are at +5 volts. Only when both inputs A and B are at -5 volts is the output Z at -5 volts.

In Figure 10A we have the two inputs that were present in Figure 8A: two square wave signals with periods of 24 ms and 12 ms with the negative going edges coinciding at 0 and 24 ms. This time, the output Z is at +5 volts when either (or both) of the inputs at A or B is at +5 volts. From 0 to 6 ms, both inputs A and B are at -5 volts; therefore, Z is also at -5 volts. From 6 to 12 ms, input A remains at -5 volts; however, input B has switched to +5 volts. As a result, output Z is also at +5 volts for this period. From 12 to 18

A	B	Z
1	1	1
0	1	1
1	0	1
0	0	0

FOR LOGIC
IN FIGURE 4
(NEGATIVE
LOGIC AS-
SIGNMENT)

A	B	Z
1	1	1
0	1	1
1	0	1
0	0	0

FOR LOGIC
IN FIGURE 6
(POSITIVE
LOGIC AS-
SIGNMENT)

A B
RESULT OF REVERSING LOGIC
ASSIGNMENTS FOR BASIC
DIODE CIRCUITS

Figure
9

DIODE LOGIC CIRCUITS

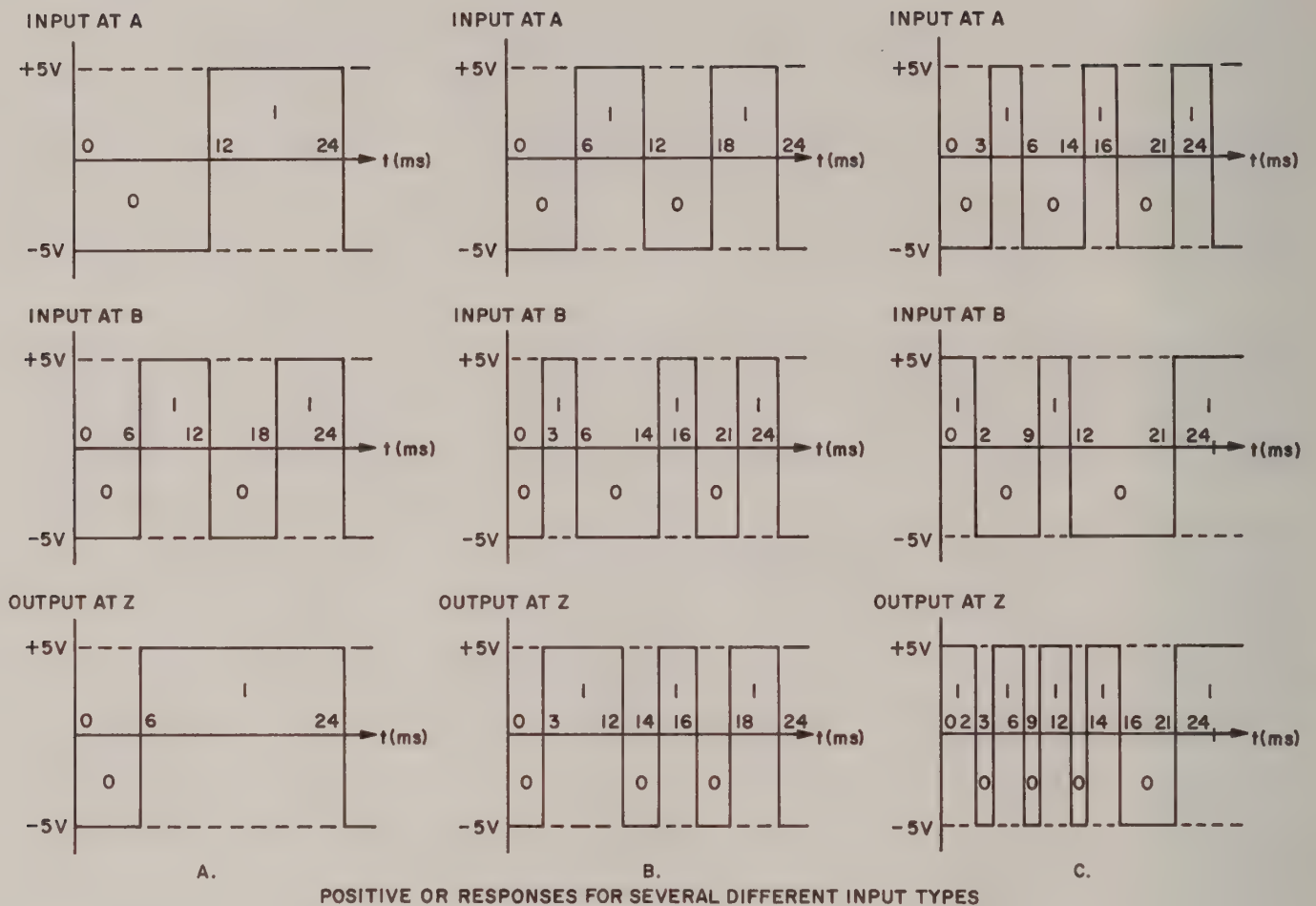


Figure 10

ms, both inputs have switched polarities. Input A has become positive while B has returned to -5 volts. The positive voltage at A means that Z will also be positive (5 volts) regardless of what B is during this time period. As in Figure 8A, the waveforms repeat after 24 ms.

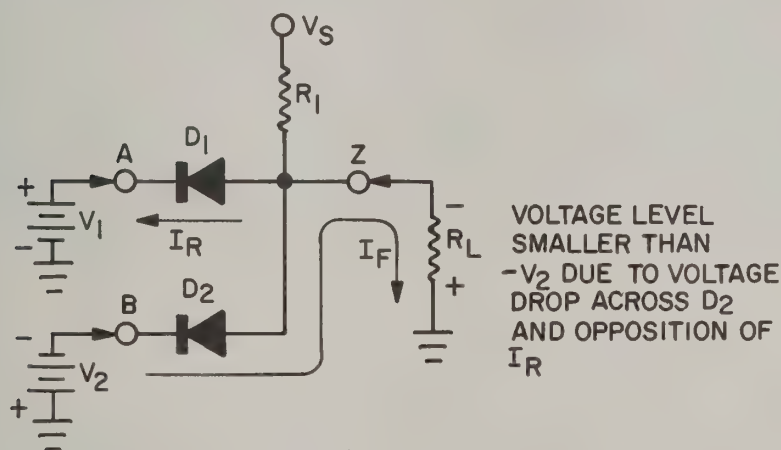
In Figure 10B the two input waveforms are the same as those in Figure 8B. Output Z is +5 volts from 3 to 12 ms because input B is +5 volts from 3 to 6 ms and input A is +5 volts from 6 to 12 ms.

Input B is also at -5 volts from 12 to 14 ms; thus, Z is -5 volts from 12 to 14 ms, +5 volts from 14 to 16 ms, and again -5 volts from 16 to 18 ms. Input A is at +5 volts from 18 to 24 ms, causing Z to return to +5 volts regardless of the input at B. Note that only between 21 and 24 ms both A and B are at +5 volts; therefore, Z is also at +5 volts.

In Figure 10C notice that output Z is +5 volts from 0 to 2 ms, 3 to 6 ms, 9 to 12 ms, 14 to 16 ms and 21 to beyond 24 ms because either or both A or B are at +5 volts.

Circuit Losses

Assuming a perfect diode, with zero forward resistance and infinite reverse resistance, the signal suffers no attenuation as it passes through the diode AND or OR gate. However, in actual practice, this is not the case. The nonideal characteristics of a diode cause signal attenuation. This limits the number of diode logic circuits which may be directly connected together (in parallel logic circuits driven by one source circuit). Likewise, the number of inputs a diode logic circuit may have is also limited by the nonideal characteristics of the diodes used.

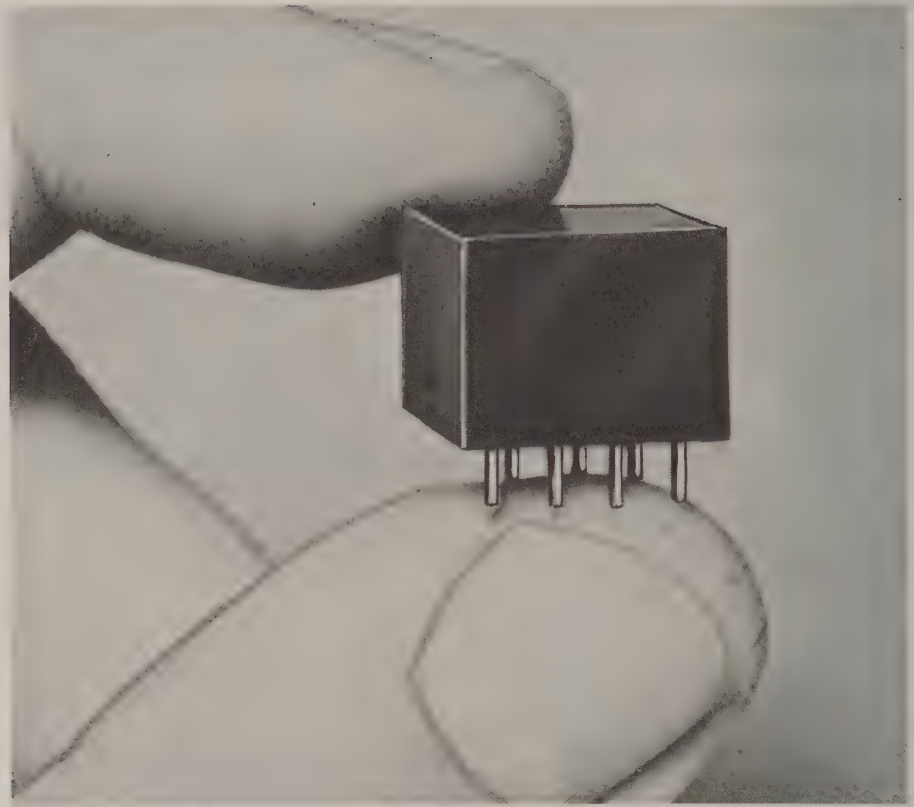


EQUIVALENT ACTION IN CIRCUIT OF FIGURE 4
WITH $A = +5$ AND $B = -5$ VOLTS

Figure 11

These limitations can be explained with the aid of an equivalent circuit. Figure 11 is an equivalent circuit of Figure 4 when input A is at +5 volts and input B is at -5 volts and the output is -5 volts. Batteries V_1 and V_2 represent inputs at A and B, respectively. D_1 is reverse biased and D_2 is forward biased. If the reverse resistance of D_1 was infinite, it may be considered to be out of the circuit. However, the reverse resistance of a practical semiconductor diode is not infinite. While usually greater than $100\text{ k}\Omega$, the actual reverse resistance depends upon the reverse bias applied to the diode.

In the equivalent circuit of Figure 11, a reverse current flowing through the load and D_1 opposes the forward current flowing through the load and D_2 .



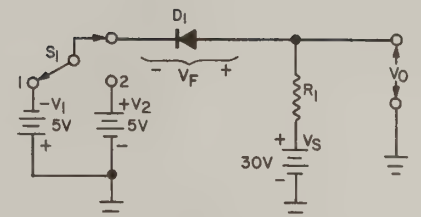
A diode-transistor logic circuit encapsulated to form a plug-in unit.

Courtesy Walkirt

In the figure, the reverse current path is represented by I_R ; the forward current path, by I_F . Although the two currents oppose each other in the load, since I_F is considerably larger than I_R , the voltage at Z will be negative—but the negative value is slightly less than it would be if D_1 were a perfect diode (allowing no reverse current).

Silicon diode reverse currents typically range from 15 to 75 microamperes; germanium diode reverse currents, from 0.6 to 1.35 milliamperes. Hence, if a diode logic circuit such as that shown in Figure 11 has a large number of inputs (and a correspondingly large number of diodes) the sum of the reverse currents for all the cut-off diodes can become large enough to produce a significant decrease in the negative voltage value at Z. In fact, it is possible for the total reverse current through R_L to completely “cancel” the forward current. For this reason, the number of inputs a practical diode gate may have is limited. Practical diode gates contain anywhere from two to twelve input diodes. They may be arranged, along with other diode gates, on a plug-in printed circuit board or they may be encapsulated in a small plug-in

module similar to a transistor case. Germanium diode gates are limited to three or four inputs since the reverse current of a germanium diode is much larger than that of a silicon diode. Diode forward resistance has an altogether different effect on circuit operation. The forward resistance prevents the output signal of a diode logic circuit from equaling the input signals. This action is demonstrated with the aid of another equivalent circuit. Figure 12 is a partial equivalent circuit for the diode gate circuit of Figure 4. S_1 and battery voltages V_1 and V_2 supply either a +5 volt or -5 volt input voltage level. Only one input, and its corresponding diode, is shown. This is all that need be considered, as circuit action is the same for any number of diodes and inputs.



PARTIAL EQUIVALENT CIRCUIT OF FIGURE 4 SHOWING EFFECT OF DIODE FORWARD RESISTANCE

Figure 12

Assume S_1 is at position 1. Under these conditions a -5 volt input voltage level is applied to the circuit. The -5 volt input voltage level, together with the +30 volt bias supply, forward biases D_1 . Assuming V_F is zero (a perfect diode), the -5 volt input signal appears across the series combination of R_1 and V_S and the output terminals. V_1 and V_S are series-aiding and the total voltage across R_1 is 35 volts. The output voltage level is the difference between V_{R_1} and V_S , which is -5 volts. However, if V_F is not zero, part of the input voltage will appear across D_1 and the rest of the input voltage, together with V_S , appears across R_1 . Thus, if V_F is 1 volt, the voltage across R_1 is reduced, and the output voltage is -4 volts.

A similar action occurs when S_1 is at position 2 and a +5 volt input voltage level is applied to the circuit. The +5 volt input voltage level at the input terminals, together with the +30 volt bias source, again forward biases D_1 . Assuming that V_F is zero (a perfect diode), the +5 volts at the input terminals appears across the series combination of R_1 and V_S . Now V_2 and V_S are series opposing and their difference, 25 volts, is across R_1 . The output voltage level is the difference between V_{R_1} and V_S which is +5 volts. However, if V_F is not zero, part of the input voltage will appear across D_1 and the rest of the input voltage, series opposing with V_S , appears across R_1 . Thus, when V_F is 1 volt, the voltage across R_1 decreases and the output voltage is +6 volts.

Circuit action may be summarized as follows. With the forward resistance (V_F) of a diode equal to zero, the output voltage has the same amplitude as the input signal. For example, with input signals of -5 volts and +5 volts, the output signals are -5 volts and +5 volts. However, if V_F , the forward resistance, is not zero the output signal levels are shifted. In this example a typical silicon diode circuit input voltage of -5 and +5 volts may produce output signals of -4 and +6 volts, respectively. To keep the shift in the output signal levels as small as possible, diodes with low forward diode drops (germanium diodes are better than silicon in this case) are used.

DIODE-TRANSISTOR LOGIC

The signal losses and level shifting associated with diode logic circuits not only limit the number of inputs to a given gate, but also prevent the direct interconnection of a large number of diode gates. However, amplifiers may be inserted between the diode logic circuits to restore the signals to their proper levels. Most often the amplifiers are built around transistors. A special name is given to the arrangement to which transistor amplifiers follow diode logic circuits. This arrangement is called **DIODE TRANSISTOR LOGIC (DTL)**.

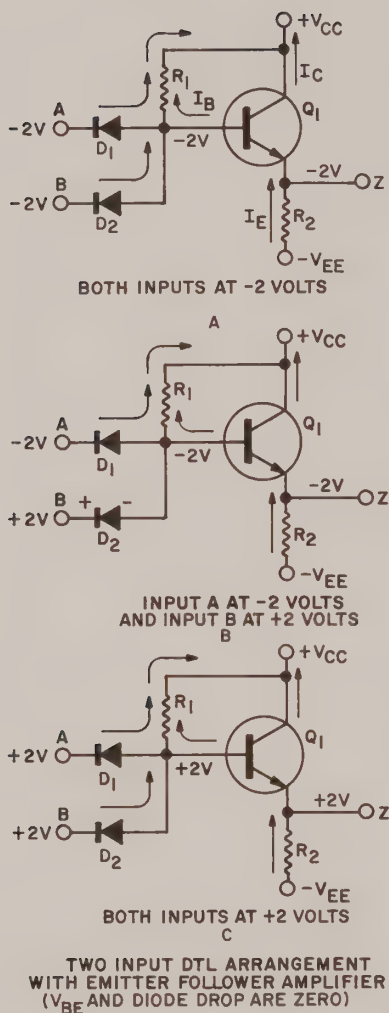


Figure 13

The common emitter configuration, and the common collector or emitter follower configuration are the most widely used in DTL circuits. Of these, the common emitter configuration has the highest voltage and power gains and develops a polarity reversal between its input and output voltage signals. As previously mentioned, the polarity of the signals in a logic circuit is very important. When the common emitter amplifier configuration is used in a DTL circuit, the polarity reversal or inversion produced by the amplifier must be taken into consideration. The common collector or emitter follower configuration, on the other hand, has a very low voltage gain (just under 1), but a high current gain, and low power gain compared to the common emitter configuration. Unlike the common emitter configuration, the common collector configuration does not develop any polarity reversal between its input and output voltage signals.

Use of Noninverting Amplifiers

Figure 13 shows the schematic diagram of a DTL circuit built around an emitter follower amplifier using an NPN transistor. Diodes D_1 and D_2 together with R_1 form a diode gate. Q_1 and R_2 form a common collector (emitter follower) type transistor amplifier. The operation of D_1 , D_2 and R_1 is as explained for the diode logic circuit of Figure 4. Assume that both inputs A and B are at -2 volts, as in Figure 13A. Under these conditions, the cathodes of D_1 and D_2 are at -2 volts, and since the anodes are connected to the positive terminal of V_{CC} , both diodes are forward biased. The currents are shown by arrows. As a result, the output of the diode gate is at -2 volts (assuming perfect diodes). In turn, this -2 volts is applied to the base of Q_1 .

The -2 volts from the diode gate makes the base of Q_1 negative with respect to ground. The emitter of Q_1 is made negative with respect to ground by $-V_{EE}$. The value of $-V_{EE}$ is made much larger than -2 volts so that, even though the base is negative with respect to ground, it is positive with respect to the emitter; therefore, the emitter junction of Q_1 is forward biased. With its emitter junction forward biased, the base-to-emitter resistance of Q_1 is very low, and Q_1 collector current is large (assuming a perfect diode). The

The following Practice Exercise questions cover the subjects which you have just studied. They are:

THE AND AND OR FUNCTION

DIODE OR GATE

CIRCUIT LOSSES

12. What is the switching function for the circuit of Figure 6 if +5 volts represents the 1 state and -5 volts represents the 0 state?
13. A diode AND gate becomes an OR gate if the logical assignments are _____.
14. When pulse signals corresponding to binary 1's are applied to all inputs, but the pulses are not of the same duration, how does the response of an AND gate differ from that of an OR gate?
15. The reverse current of a cutoff diode in a diode logic circuit opposes the desired output voltage. True or False?
16. Since the reverse current for a diode is very small, how can this current limit the number of inputs associated with a diode gate?
17. As shown by the equivalent circuit of Figure 12, the output voltage levels are shifted by the (a) forward resistance of D_1 , (b) reverse resistance of D_1 .

18. The drop across the diode in the figure below is 2 volts when it is conducting. If an input signal is -10 volts, the output signal will be _____, and if an input signal is $+10$ volts, the output signal will be _____.



-2 volts at the base of Q_1 is coupled to the emitter (assuming a perfect transistor). As a result, the Q_1 emitter-to-ground voltage (output Z) is -2 volts.

Now assume that input A is at -2 volts and input B is at $+2$ volts (Figure 13B). Under these conditions, D_1 remains forward biased; and therefore D_1 couples the -2 volts at input A to the base of Q_1 and the anode of D_2 . The -2 volts at the base of Q_1 forward biases Q_1 , and output Z, the Q_1 emitter-to-ground voltage, becomes -2 volts. This time only D_1 conducts current (as shown by the arrows), and D_2 is reverse biased with 4 volts across D_2 (as shown by the plus and minus signs). The action is the same if input A is at $+2$ volts and input B is at -2 volts. The only difference is that D_1 is now reverse biased and D_2 is forward biased. Output Z remains at -2 volts.

Finally, assume that both inputs A and B are at $+2$ volts (Figure 13C). Then the cathodes of D_1 and D_2 are at $+2$ volts with respect to ground. Their anodes are connected to the positive terminal of V_{CC} through R_1 . Since V_{CC} is much larger than $+2$ volts, the anodes of D_1 and D_2 are positive with respect to their cathodes, and D_1 and D_2 are forward biased. Again, current flows through both diodes because they are both forward biased, and this current, along with I_B , flows through R_1 to V_{CC} . The $+2$ volts at both input terminals is coupled to the base of Q_1 . Together with $-V_{EE}$ in the emitter circuit, the $+2$ volts at the base of Q_1 forward biases the emitter junction of Q_1 . As a result, the base-to-emitter voltage of Q_1 is zero. Therefore, when both inputs A and B are at $+2$ volts, output Z is at $+2$ volts.

Combining an emitter follower amplifier with a diode logic circuit to form a DTL circuit has many advantages. One important advantage is that the emitter follower does not reverse the polarities of the 0 and 1 signal levels. Therefore, the logical operation performed by the diode gate circuit is not altered by the emitter follower amplifier. Another advantage of the emitter follower is its power gain. Although the input and output voltage signals of the emitter follower are almost the same magnitude, the output current can be much greater than the input current (by a factor of β). An emitter follower amplifier can have a current gain in excess of 30. As a result, a single DTL circuit with relatively low power signals at its input can drive many additional logic circuits.

The switching time of an emitter follower amplifier is also very short. In the circuit of Figure 13, Q_1 always conducts. The input signals never drive Q_1 into cutoff or saturation. With this arrangement any current carrier storage effects that occur in Q_1 are very small. The overall switching time of the diode transistor logic circuit with common collector amplifiers depends largely upon the switching time of the diode logic circuit.

The DTL circuit of Figure 13 uses an NPN type transistor. A PNP transistor can also be used if the polarities of the bias batteries are changed to pro-

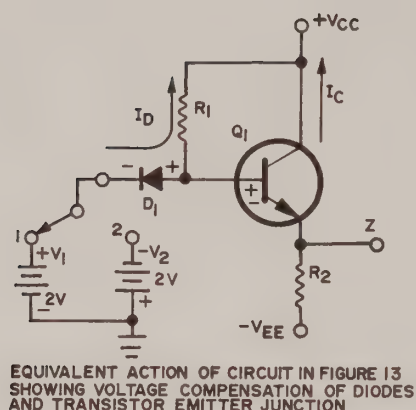


Figure
14

vide proper bias for the PNP transistor. An emitter follower amplifier may also be connected to the diode gate of Figure 6. Again, either PNP or NPN transistors can be used if the polarities of the bias batteries provide proper bias for the transistor used.

In the previous explanation of Figure 13, it was assumed that the forward resistances of the diodes were zero. In addition, it was also assumed that the emitter junction of the transistor had zero internal resistance when forward biased. However, in actual circuits it is impossible to obtain zero resistance across forward biased junctions. Both the diodes and the emitter junction of the transistor have definite internal resistances when forward biased. In addition, the voltage gain of an emitter follower amplifier is always a little less than 1. Thus, it would seem that the addition of the emitter follower amplifier would decrease the amplitude of the output signal from the diode gate. However, this is not the case. The voltages across the internal resistances of the diodes and the voltage across the emitter junction of the transistor compensate for each other if their drops are the same.

To show this action the circuit of Figure 13 has been simplified in Figure 14. Only one silicon diode is shown. The action is the same regardless of the number of diodes. Batteries V_1 and V_2 , together with S_1 , provide either a +2 volt or a -2 volt input. Let's assume S_1 is at position 1, applying +2 volts to the circuit. Since V_{CC} is greater than the input voltage, D_1 is forward biased and a current shown by the I_D arrow is developed. This current develops a voltage of the indicated polarity across the forward resistance of D_1 .

Notice that this voltage is series-aiding with the input voltage, and their sum is the base-to-ground voltage of Q_1 . For illustrative purposes, assume the voltage across D_1 is 0.5 volt. Thus, the Q_1 base-to-ground voltage is the input voltage (+2 volts) plus the voltage across D_1 (0.5 volt), which equals +2.5 volts. The +2.5 volts at the base of Q_1 together with $-V_{EE}$ forward biases the emitter junction of Q_1 . In turn, collector current produces a voltage of the polarity indicated across the Q_1 emitter junction. This voltage is series-opposing the base-to-ground voltage of Q_1 ; therefore, the emitter-to-ground voltage of Q_1 is the difference between the base-to-ground voltage and the voltage across the emitter junction.

Assume that the voltage across the emitter junction is 0.5 volt. The emitter-to-ground voltage is the difference between the base-to-ground voltage (2.5 volts) and the voltage across the emitter junction (0.5 volt), which equals +2 volts. Note that this voltage is exactly the same as the input voltage. **NOTE THAT NOW THE VOLTAGE ACROSS THE DIODE AND THE EMITTER JUNCTION OF Q_1 ARE OPPOSITE AND HAVE CANCELLED EACH OTHER.**

Circuit action is the same if S_1 is moved to position 2, applying -2 volts to the circuit. The -2 volts at the input forward biases D_1 . This results in current I_D and a voltage of the indicated polarity is developed across the forward resistance of D_1 . Again assume that the voltage across D_1 is $.5$ volt. The voltage across D_1 (0.5 volt) and the input voltage (-2 volts) are series-opposing, and the difference between them (-1.5 volts) is applied to the base of Q_1 .

The -1.5 volts from base-to-ground, together with $-V_{EE}$, forward biases Q_1 . Collector current, indicated by the I_C arrow, produces a voltage of the polarity shown across the emitter junction of Q_1 . This voltage is series-aiding with the base-to-ground voltage so that the emitter-to-ground voltage is the sum of the base-to-ground voltage and the voltage across the emitter junction. Again assume the voltage across the emitter junction resistance is 0.5 volt. The emitter-to-ground voltage of Q_1 is the sum of the base-to-emitter voltage (-0.5 volt) and the base-to-ground voltage (-1.5 volts), or -2 volts. Note again that this voltage is the same as the voltage applied to the input terminals. The voltage across the forward resistance of the diode and the voltage across the emitter junction are of opposite polarity and cancel each other. By properly choosing diodes and transistors, the circuit can be so arranged that minimum signal attenuation occurs.

However, the reverse current of the diodes and the gain of the CC circuit being a little less than one both contribute to the loss of signal swing as it goes through a noninverting DTL OR or AND gate.

DTL circuitry may be mounted directly on printed circuit boards, as described for diode logic. However, more frequently you will find DTL modules. These units plug into printed circuit boards and can easily be replaced.

Use of Inverting Amplifiers

The diode transistor logic (DTL) circuits previously described suffer from one disadvantage. An emitter follower (common collector) type of transistor amplifier produces a voltage gain of less than one (and therefore cannot make up for circuit losses). The output voltage swing of a DTL gate using an emitter follower amplifier is a little less than the input voltage swing. Both the leakage of the diodes and the voltage gain of a little under unity for the CC stage contribute to the reduction of voltage swing from a noninverting DTL gate. In fact, the loss in voltage swing is complicated when such gates are cascaded. For example, a CC DTL gate whose output swing is 90% of the input swing has a 27% reduction in voltage swing when three stages are cascaded $[100\%(1 - .9^3) = 100\%(1 - .73) = 27\%]$.

Where it is desired that the voltage swings be maintained, DTL gates with CE amplifiers are usually employed instead of the emitter follower. It is possible to alternate CE and CC gates in DTL circuitry because a CE stage can easily make up the loss in voltage swing in a CC stage or two.

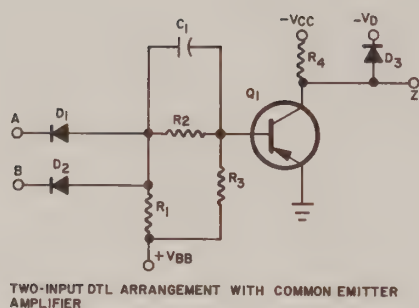


Figure 15

Figure 15 shows the diode gate circuit of Figure 4 connected to a common emitter type amplifier. Diodes D_1 and D_2 and resistor R_1 form the diode gate circuit. The remaining components make up a PNP common emitter type amplifier. The output voltage from this diode gate is applied to the base of Q_1 through the input coupling network composed of C_1 and R_2 . This network helps to decrease the circuit switching time. In addition to supplying bias for the diode logic circuit, V_{BB} supplies fixed bias for Q_1 through R_3 . The output signal at Z is taken from the collector of Q_1 to ground. R_4 is the collector load resistor and $-V_{CC}$ is the collector supply. To prevent saturation of Q_1 , a clamping circuit consisting of D_3 and $-V_D$ is included. This circuit prevents Q_1 collector voltage from decreasing to the point where Q_1 goes into saturation. Unsaturated operation is desired since the switching time of an unsaturated circuit is much less than that of a saturated one.

To analyze the circuit operation, assume that both inputs A and B are at +2 volts. V_{BB} is sufficiently large that both D_1 and D_2 conduct, and a positive voltage appears at the output of the diode logic circuit. This positive voltage is applied to the base of Q_1 through C_1 and R_2 . The positive voltage aids the reverse bias set up at the emitter junction of Q_1 by R_3 and V_{BB} , driving Q_1 far into cutoff. When Q_1 is cut off, collector current is low and the voltage across the collector load resistor R_4 is very small compared to $-V_{CC}$. Since the output at Z is the difference between V_{R_4} and $-V_{CC}$, the output at Z is approximately equal to $-V_{CC}$. Note the polarity reversal developed by the Q_1 amplifier. The input signal applied to the base is positive and output signal at Z (the Q_1 collector voltage) is negative. With the Q_1 collector voltage highly negative, D_3 remains reversed biased since $-V_D$ is much lower than $-V_{CC}$.

Now assume that inputs to the A and B are both -2 volts. Under these conditions D_1 and D_2 conduct and a negative voltage is coupled to the base of Q_1 through C_1 and R_2 . This negative voltage opposes the reverse bias set up by R_3 and V_{BB} . As a result, Q_1 is driven out of cutoff and toward saturation. The voltage across R_4 due to I_C subtracts from $-V_{CC}$, and the Q_1 collector voltage at Z, which is the output, decreases.

When the Q_1 collector voltage becomes less than $-V_D$, diode D_3 becomes forward biased and the collector of Q_1 is connected through the low forward resistance of D_3 to the negative terminal of $-V_D$. Then, no matter how hard the transistor tries to go into saturation, and increase the voltage across R_4 , the collector voltage of Q_1 never decreases more than a diode drop below the value of $-V_D$. The value of $-V_D$ is chosen so that it maintains the Q_1 collector voltage just above the saturation point. As a result, Q_1 operates as

an unsaturated switch and therefore its switching time is very short. The value of $-V_D$ is usually 2 to 3 volts in circuits of this type.

Circuit action is the same if input A is at +2 volts and input B is at -2 volts. In this case D_2 is forward biased and a negative voltage is applied to the base of Q_1 through the C_1R_2 network. D_1 , on the other hand, is reverse biased. With a negative voltage at its base, Q_1 conducts collector current and the Q_1 collector voltage, output at Z, decreases to about the value of $-V_D$. The action is similar if input A is negative and input B is positive. In this case D_1 couples a negative voltage to the base of Q_1 , and D_2 is cut off.

The polarity reversal produced by Q_1 alters the logical operation of the diode gate. When inputs A and B are at +2 volts, D_1 and D_2 conduct. In turn, Q_1 is driven into cutoff and the output at Z is approximately equal to $-V_{CC}$. However, when either or both inputs A and B are at -2 volts, D_1 and D_2 again conduct. Now Q_1 is driven into conduction and the output at Z is equal to about $-V_D$, which is very small. The input and resultant output voltage polarity for the circuit of Figure 15 are shown in truth table form in Figure 16A. In the Z column, the + signs represent the condition when output or Q_1 collector voltage is equal to $-V_D$. At the inputs, a minus sign indicates a negative voltage input and the plus sign a positive input voltage.

A	B	Z
-	-	+
+	-	+
-	+	+
+	+	-

A

A	B	Z
0	0	1
1	0	1
0	1	1
1	1	0

B

TRUTH TABLE CONSTRUCTION FOR CIRCUIT IN FIGURE 15 WITH POSITIVE LOGIC ASSIGNMENT

Figure 16

To examine the logical operation performed by the circuit, binary digits 0 and 1 are chosen and substituted into the truth table of Figure 16A. Assume that a 0 is a negative voltage and a 1 is a near zero voltage output on a positive input. Now all the negative signs in the truth table of Figure 16A can be replaced with 0's and all the + signs with 1's. This results in the truth table of Figure 16B. Examining this table shows that under these signal conditions, the circuit of Figure 15 is a NAND (NOT-AND) gate. When both inputs at A and B are 1, output at Z is 0; and when either or both inputs at A and B are 0, output at Z is 1. Since the circuit performs the NAND function when the 1 is a near zero output voltage or a positive input voltage, the circuit is a positive NAND gate.

You may recall that the NAND function:

$$\overline{A \cdot B} = Z \quad (\text{NAND}) \quad (3)$$

or:

$$A \cdot B = \overline{Z} \quad (\text{NAND}) \quad (3a)$$

is the opposite, or complement, of the AND function. That is, the AND output is 1 (see Table 1) only when all inputs are 1; conversely, the NAND function output is 0 only when all inputs are 1.

Diode logic and DTL circuits with noninverting amplifiers cannot provide the NAND operation. Reversing the 0 and 1 signals changes the logical

A	B	Z
-	-	+
+	-	+
-	+	+
+	+	-

A

A	B	Z
1	1	0
0	1	0
1	0	0
0	0	1

B

TRUTH TABLE CONSTRUCTION
FOR CIRCUIT IN FIGURE 15 WITH
NEGATIVE LOGIC ASSIGNMENT

Figure
17

operation of the circuit. Now assume that the 1 is a negative voltage and the 0 is a positive or near zero voltage. Substituting 0's and 1's into the truth table of Figure 16A (repeated as Figure 17A) produces the truth table of Figure 17B. **EXAMINING THIS TRUTH TABLE SHOWS THAT WHEN 0 IS A POSITIVE VOLTAGE AND 1 A NEGATIVE VOLTAGE, THE CIRCUIT OF FIGURE 15 IS A NOR (NOT-OR) GATE.** When either or both inputs at A and B are 1, the output at Z is 0; and when both inputs at A and B are 0, the output at Z is 1. Since the circuit performs the NOR function when the 1 is a negative voltage, the circuit is also a negative NOR gate. The circuit of Figure 15 is shown using a PNP transistor. If desired, an NPN transistor can be used; however, the polarities of all of the bias supplies must be changed to provide proper bias for and operation of the NPN transistor. The NOR function:

$$\overline{A + B} = Z \quad (\text{NOR}) \quad (4)$$

or:

$$A + B = \overline{Z} \quad (\text{NOR}) \quad (4a)$$

is the complement of the OR function. That is, while the OR output is 0 (see Table 2) only when all inputs are 0, the NOR output is 1 only when all inputs are 0. Diode logic and DTL circuits with noninverting amplifiers cannot provide the NOR operation.

In the circuit of Figure 15, Q_1 operates as an unsaturated switch. Saturation of Q_1 is prevented by limiting the minimum Q_1 collector voltage to the value of $-V_D$. In some cases it may be desirable to operate Q_1 as a saturated switch. Although this increases circuit switching time, the increase in switching time is compensated for by the reduced power dissipation of Q_1 . The power dissipated by a transistor operated as a saturated switch is much lower than when the transistor is operated as an unsaturated switch. Q_1 in Figure 15 can be operated as a saturated switch by simply removing the $V_D D_3$ circuit. Then, when Q_1 is driven into conduction, Q_1 collector voltage can decrease to the saturation level.

The advantage of using a common emitter amplifier following a diode logic circuit, as in Figure 15, is the voltage gain produced by Q_1 . Very small input signals applied to the diode logic circuit produce large output signals. However, it is important to note that the polarity reversal produced by a grounded emitter amplifier reverses the 0 and 1 logic levels. If this reversal is not desired, another common emitter amplifier can be used to restore the signals to their original polarities.

Low Level Logic

In the DTL circuit of Figure 15, large input voltages are applied to the diode gate circuits, and the outputs of the diode gates are applied to the transistor

amplifier. The large output voltages from the diode gates are applied to the base of the transistor through current-limiting resistor R_2 . When the input signal level changes, speed-up capacitor C_1 causes a very large signal to be applied to the base to ensure a shorter circuit switching time.

Instead of using large input signals applied through a current-limiting resistor, small input signals may be applied directly to the base of the transistor. This eliminates the need for a current-limiting resistor and a speed-up capacitor.

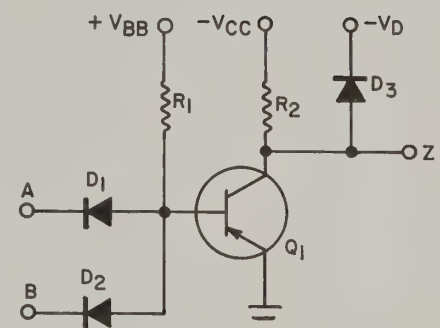
Only a few tenths of a volt change (applied to the base-to-emitter junction) is needed to drive a transistor from saturation to cutoff. This arrangement is called **LOW LEVEL LOGIC** because the signal levels used are very small.

A typical low level logic DTL circuit is shown in Figure 18. Note the similarity between the circuit of Figure 18 and the conventional DTL circuit of Figure 15. The only difference is that the input coupling network, C_1 , R_2 , and R_3 in Figure 15, is not used in Figure 18. Both circuits operate similarly except that the signal levels applied to the circuit of Figure 18 are much smaller than those applied to the circuit of Figure 15. Signal levels of +0.3 volt and -0.3 volt are typical for the circuit of Figure 18. Compare this to the +5 and -5 volt signals which are typical for DTL circuits such as the one of Figure 15.

Low level logic circuits are sometimes referred to as **CURRENT SWITCHED CIRCUITS** because the input or base current changes are greater than the base voltage changes.

There are certain advantages in using small signal levels. With small signal levels, little current is used in charging circuit capacitance. This results in small current losses and improved circuit operation. Some improvements are: faster switching, lower power dissipation, and more gates can be driven from a single source. In addition, the sources or other gate circuits that are driving the low level logic DTL circuit need only deliver small output voltage levels.

Stability in low level logic DTL circuits is a problem. The small signal levels must be carefully controlled to insure proper circuit operation. Temperature increases affect stability also. Furthermore, the use of small signal levels makes this type of circuit more susceptible to noise pulses. Stray signals can easily activate a low level logic DTL circuit unless the circuit is properly shielded.



LOW LEVEL DTL CIRCUIT

Figure
18

TUNNEL DIODE SWITCHING

Extremely high speed switching circuits can be constructed with tunnel diodes (almost all of which are made of germanium). The tunnel diode is similar to a conventional semiconductor diode in that both types of diodes are built around a PN junction. However, the P and N materials used in constructing tunnel diodes contain more impurities than the materials used to construct conventional diodes. The greater concentration of impurities causes the internal resistance of a tunnel diode to be much less than that of a conventional diode. As a result, the characteristics of a tunnel diode are quite different from those of a conventional diode.

Figure 19 shows the forward characteristics of a typical tunnel diode and a conventional semiconductor diode. The forward characteristics of a conventional semiconductor diode are shown by the heavy broken line. The tunnel diode characteristics are shown by the heavy solid line. Notice the

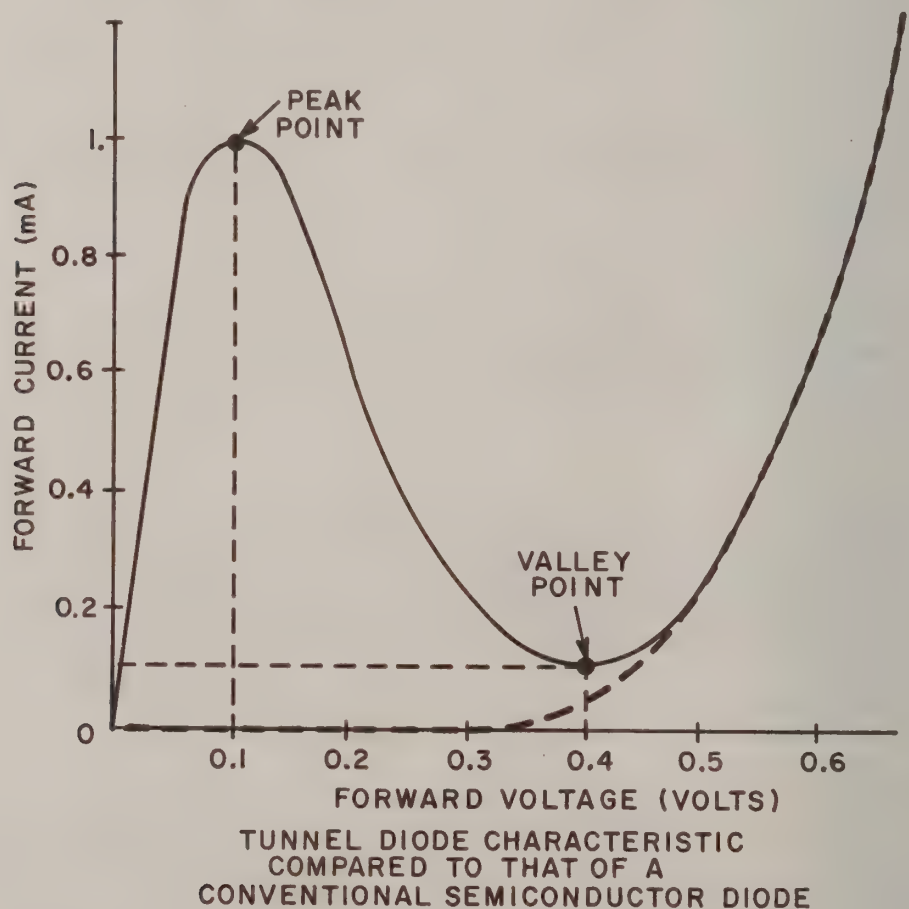


Figure 19

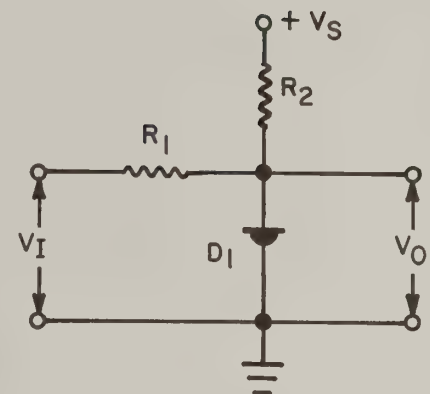
difference in the two characteristic curves. Forward current does not flow in the conventional diode until the forward voltage is increased beyond 0.35 volt. However, in the tunnel diode, forward current increases, decreases, and starts to increase again in the region between 0 and 0.4 volt.

The region between the peak point and the valley point is very important. In this region, as the forward voltage is increased, forward current decreases. The tunnel diode has a negative internal resistance in this region. Outside the peak to valley point region, diode current increases as the forward voltage is increased. The internal resistance is positive in this region.

The negative resistance characteristics of a tunnel diode may be used to construct very high speed switching circuits. A basic tunnel diode switching circuit is shown in Figure 20. V_S forward biases tunnel diode D_1 . The input signal V_I is applied through current-limiting resistor R_1 across D_1 . R_2 acts as a load resistor and the output signal V_O is taken across D_1 . The characteristic curve of D_1 is shown in Figure 21. The ac load line shown on the curve represents the points at which D_1 in Figure 20 operates. The ac load line is determined by the values of R_2 and V_S . The tunnel diode will operate at some point on this load line, depending upon the input signal applied to the circuit.

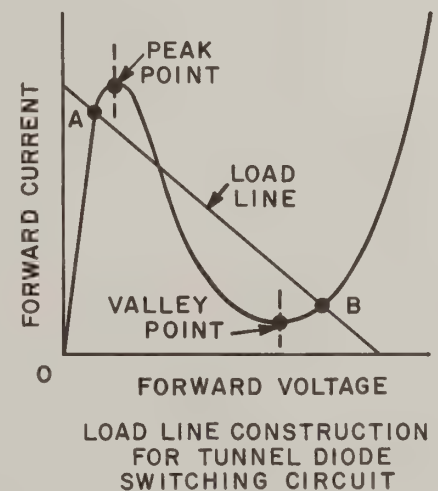
With no signal applied, D_1 operates at point B on its characteristic curve. This point is outside the negative resistance region of the diode (peak point to valley point). The diode will remain at this operating point until an input signal of the proper polarity is applied to shift the diode to point A. At point B, the diode forward current is low and the voltage across R_2 is small. As a result, the voltage across D_1 (the output voltage) V_O , is maximum.

Now assume that a negative signal is applied to the input terminals. This negative signal applied through R_1 across D_1 opposes the forward bias applied to D_1 by $+V_S$. As a result, the forward voltage across D_1 decreases. If the forward voltage becomes less than the voltage at the valley point, diode current increases. In turn, the voltage across R_2 increases, further reducing the forward voltage across D_1 . The decrease in forward voltage across R_2 causes another increase in diode current which further reduces the forward voltage across D_1 . This regenerative action continues until the diode operating point has switched to A. The diode operates at point A (the "on" condition, where the forward voltage is about 0.05 volt) until an input signal of the proper polarity is applied to the input terminals to switch it back to point B (the "off" condition where the forward voltage is about 0.5 volt). At point A, diode current is maximum and the voltage across R_2 is maximum. As a result, the forward voltage across D_1 (the output voltage) is very low (close to zero). The forward voltage swing is about 0.5 volt for germanium tunnel diodes. With D_1 operating at point A, assume that a positive signal is applied to the input terminals. The positive input signal



BASIC TUNNEL DIODE SWITCHING CIRCUIT

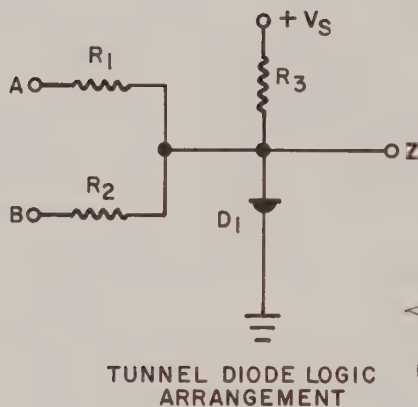
Figure 20



LOAD LINE CONSTRUCTION FOR TUNNEL DIODE SWITCHING CIRCUIT

Figure 21

applied through R_1 , across D_1 aids the forward bias applied to D_1 by $+V_S$. If the forward voltage developed by the input signal is large enough to drive D_1 past the peak point, a decrease in diode current occurs. A decrease in diode current causes a decrease in the voltage across R_2 , thereby increasing the forward voltage across D_1 . The increased forward voltage causes another reduction in diode current and a further increase in the forward voltage across D_1 . This regenerative action continues until the diode operating point has switched back to B. At point B diode current is low and the forward voltage across D_1 (the output voltage) is high. The diode operates at this point until a sufficiently negative input signal is applied to the input terminals to cause the diode to switch to point A on the load line.



Due to the regenerative switching action, extremely short switching times are possible. Tunnel diode switching circuits which can change operating states in 1 nanosecond have been developed. Because of their high switching speed, tunnel diodes are desirable in many digital systems.

Besides acting as a switch, the tunnel diode switching circuit of Figure 20 acts as a bistable circuit. Once the circuit is switched, it remains in the particular stable state until another input signal, of opposite polarity, is applied to change the circuit to its other operating state. This operation is similar to that of a flip-flop.

Figure
22

AND and OR logical operations can be performed with the tunnel diode circuit of Figure 20 if additional inputs are added. Figure 22 shows such an arrangement. The circuit can be used as an AND or OR gate, depending upon the amplitudes of the input signals and the values of R_1 and R_2 . The circuit operates as an AND gate if the amplitudes of the input signals and the values of R_1 and R_2 are such that the circuit changes states only when both input signals are applied. The circuit operates as an OR gate if the amplitudes of the input signals and the values of R_1 and R_2 are such that the circuit changes states when either or both input signals are applied.

Tunnel diode circuits do have many of the problems that low level logic circuits have: instability with temperature changes and susceptibility to noise and stray signals.

SUMMARY

Semiconductors are widely used in logic circuits because of their small size, low heat dissipation, and high reliability over a long operating life. AND and OR gates can be built around diodes. A NOT circuit consists of a voltage amplifier that produces signal inversion. The NOT circuit cannot be built with conventional diodes alone.

The following Practice Exercise questions cover the subjects which you have just studied. They are:

DIODE-TRANSISTOR LOGIC

USE OF NONINVERTING AMPLIFIERS

USE OF INVERTING AMPLIFIERS

LOW LEVEL LOGIC

TUNNEL DIODE SWITCHING

19. No polarity reversal between input and output signals is developed by the Q_1 amplifier stage of Figure 13. True or False?
20. Q_1 in Figure 13 is usually driven into saturation by the input signals so that short switching times can be obtained. True or False?
21. The use of an emitter follower with diode logic circuitry (a) affects the logical value assignments or the logical functions, (b) does not affect the logical value assignments or the logical functions.
22. Why is it possible for an emitter follower, with a voltage gain of less than unity, to correct for the level-shifting action of forward biased diodes?
23. Q_1 in the DTL circuit of Figure 15 operates as a common emitter amplifier. True or False?
24. The input coupling network C_1 and R_2 in Figure 15 helps to reduce circuit switching time. True or False?
25. What is the purpose of D_3 and $-V_D$ in the DTL circuit of Figure 15?
26. The circuit of Figure 15 produces a function that is the _____ of the AND or OR functions because of the _____ associated with Q_1 .

27. NOR and NAND functions cannot be obtained with diode logic or non-inverting DTL circuitry. True or False?
28. Why can the circuit of Figure 18 be switched with extremely small input voltages?
29. What are two problems frequently encountered in low level logic circuitry?
30. What tunnel diode region is essential to the development of high-speed switching circuits?
31. What quantities control the function of the tunnel diode gate in Figure 22?

The same basic diode circuit can be used for both the AND and OR operations. In fact, a circuit arranged to give positive logic AND operation will also provide negative logic OR operation. For the positive logic AND circuit, a logical 1 is associated with more positive or less negative signals. For the negative logic OR circuit, a logical 1 is associated with more negative or less positive signals. Reversing the diode orientation and bias polarity in this circuit will provide the positive logic OR operation or the negative logic AND operation—again depending on the logic assignments made.

Although diode logic circuits offer the advantage of simplicity and low cost, they provide no signal amplification. Due to the imperfect characteristics of a semiconductor diode, a diode logic circuit shifts the levels and reduces the voltage swing of the signals. This prevents the direct interconnection of a large number of diode logic circuits. Amplifiers must be inserted between the diode logic circuits to restore the signals to their proper levels. The number of inputs to a diode logic circuit is also limited by the reverse current passed by reverse biased diodes. If a large number of inputs are required, diodes with very low reverse current characteristics must be used. The forward resistances of the diodes in the circuit cause the output signal levels to be shifted from those of the input signal.

To overcome the disadvantages which exist in diode logic circuits, transistor amplifiers are added to the circuit. This arrangement is called diode transistor logic, abbreviated as DTL. The transistor amplifier not only restores the signals to the desired levels, but also provides a higher power output signal. This enables the logic circuit to drive a large number of loads.

Common collector (emitter follower) amplifiers are used when no signal inversion or polarity reversal in the amplifier is desired. In addition, a common collector amplifier produces a large power gain although its voltage gain is slightly less than one. If signal inversion is desired, a common emitter type transistor amplifier is used. Besides providing signal inversion, the common emitter type amplifier has a high voltage gain. The use of a common emitter amplifier transforms diode AND or OR gates into NAND or NOR gates, respectively.

Tunnel diodes are used to construct extremely high speed switching circuits. Due to the regenerative action, characteristic of tunnel diodes, these diode switching circuits have been devised which can change states in as little as 1 nanosecond (10^{-9} seconds).

IMPORTANT DEFINITIONS

CURRENT SWITCHED CIRCUITS—See LOW LEVEL LOGIC.

DIODE LOGIC (DL)—A logic circuit arrangement which uses diodes to perform logical operations. The output of DL is not inverted and requires amplifiers to maintain the correct logic level through several logic circuits connected in series.

DIODE TRANSISTOR LOGIC (DTL)—A logic circuit arrangement which uses semiconductor diodes to perform the logic operation and a transistor amplifier is used to amplify the output signal from the diode logic circuit. The output is usually inverted.

DYNAMIC LOGIC—Logic arrangements which use short-duration pulses or the absence of pulses to represent the binary digits 0 and 1.

LOW LEVEL LOGIC—A DIODE TRANSISTOR LOGIC arrangement which can be switched by extremely low input voltage signals.

NEGATIVE AND GATE—A logic arrangement which produces a low-level output voltage signal only when all input voltage signals are at a low level.

NEGATIVE OR GATE—A logic arrangement which produces a low level output voltage signal when any one (or more) of the input voltage signals is at a low level.

POSITIVE AND GATE—A logic arrangement which produces a high-level output voltage signal only when all input voltage signals are at a high level.

POSITIVE OR GATE—A logic arrangement which produces a high level output voltage signal when any one (or more) of the input voltage signals is at a high level.

STATIC LOGIC—Logic arrangements which use long-duration voltage levels to represent the binary digits 0 and 1.

ESSENTIAL SYMBOLS AND EQUATIONS

A, B, C, D Arbitrarily selected input variables for logical switching circuits

X, Y, Z Arbitrarily selected output variables for logical switching circuits

$$Z = A \cdot B \quad (\text{AND}) \quad (1)$$

$$Z = A + B \quad (\text{OR}) \quad (2)$$

$$\overline{A \cdot B} = Z \quad (\text{NAND}) \quad (3)$$

$$A \cdot B = \overline{Z} \quad (\text{NAND}) \quad (3a)$$

$$\overline{A + B} = Z \quad (\text{NOR}) \quad (4)$$

$$A + B = \overline{Z} \quad (\text{NOR}) \quad (4a)$$

PRACTICE EXERCISE SOLUTIONS

1. AND and OR

$$2. Z = A \cdot \bar{B}$$

$$3. \bar{Z} = \bar{A} \cdot \bar{B} \cdot \bar{C}$$

$$4. Z = A + \bar{B}$$

$$5. Z = A \cdot B \cdot C \cdot D$$

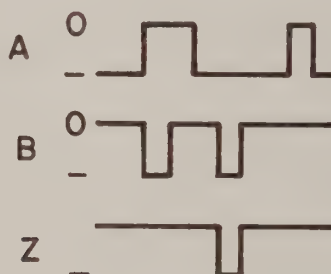
6. -5 volts

7. +5 volts

8. polarity

9. Additional inputs can be added by adding more diodes. One diode is required for each input.

10. The circuit is a negative AND gate (see Figure 6), resulting in the output shown below.



11. -5 volts

12. OR

13. reversed

14. When simultaneous "1" pulse inputs are not of the same duration, the AND gate "1" output corresponds to the shortest pulse; the OR gate "1" output, to the longest pulse.

15. True

16. Each cutoff diode allows a small reverse current to flow through the load, in opposition to any forward current, producing the desired output voltage level. However, since the cutoff diode currents are additive through

PRACTICE EXERCISE SOLUTIONS (Continued)

the load, each additional diode causes a reduction in the effective output voltage. Eventually this effect becomes significant and can even reduce the output voltage to zero.

17. (a) forward resistance of D_1
18. -12 volts; +8 volts
19. True
20. False— Q_1 is never driven into saturation in this circuit arrangement. It always operates within its active region.
21. (b) does not affect the logical value assignments or the logical functions.
22. The circuit arrangement is such that the diode voltages which tend to shift the signal level are exactly cancelled by the emitter junction voltage of the transistor if the junction drops are equal.
23. True
24. True
25. D_3 and $-V_D$ form a clamping circuit used to prevent saturation of Q_1 . This eliminates the transistor storage effects from the circuit switching time characteristics.
26. complement, signal inversion
27. True
28. Small voltages are sufficient because there are no current-limiting resistors in the base-emitter circuit of Q_1 .
29. Circuit stability and undesirable switching due to stray noise are two problems encountered in low level logic circuitry.
30. The negative resistance region of the tunnel diode is essential to high-speed switching applications.
31. The relative values of the input signals and the resistors R_1 and R_2 control the AND and OR operation of the tunnel diode gate.

1. B - THE DIODE GATE CIRCUIT OF FIGURE 4 DEVELOPS A POSITIVE OUTPUT SIGNAL, Z, WHEN -- BOTH INPUTS A AND B ARE POSITIVE.

If the input at A and/or B is negative, diode D_1 and/or D_2 will conduct and pass the negative voltage to the output at Z. However, if both inputs at A and B are positive, diodes A and B conduct and pass the positive voltage to the output at Z.

2. B - IF THE 0 SIGNAL IS +5 VOLTS AND THE 1 SIGNAL IS -5 VOLTS, THE DIODE GATE OF FIGURE 4 IS -- AN OR GATE.

The output at Z will be a logical 1 (-5 volts) when the inputs at A and/or B are a logical 1. This type of logical decision is the characteristic of an OR gate.

3. C - IN THE DIODE GATE CIRCUIT OF FIGURE 6, WHEN INPUT A IS POSITIVE AND INPUT B IS NEGATIVE, -- D_1 IS CONDUCTING AND D_2 IS CUT OFF.

The positive voltage at input A, together with the -30 volt bias source V_S , forward biases D_1 . With D_1 forward biased, its internal resistance is zero (assuming a perfect diode) and the positive voltage at input A is applied through D_1 to output Z. The positive voltage at output Z is also applied to the cathode of D_2 . The negative voltage at input B together with the positive voltage at output Z reverse biases D_2 .

4. D - IF THE 0 SIGNAL IS +5 VOLTS AND THE 1 SIGNAL IS -5 VOLTS, THE DIODE GATE OF FIGURE 6 IS -- AN AND GATE.

The output at Z will be a logical 1 (-5 volts) only when both the inputs at A and B are a logical 1.

5. A - CIRCUIT CAPACITANCE C_1 IN FIGURE 15 -- REDUCES CIRCUIT SWITCHING TIME.

When the input signal level changes, capacitor C_1 causes a very large signal to be applied to the base of transistor Q_1 , thus ensuring a shorter switching time.

6. A - THE TRANSISTOR AMPLIFIER USED IN THE DTL CIRCUIT OF FIGURE 13 -- IS AN EMITTER FOLLOWER TYPE AMPLIFIER.

In common collector (emitter follower) amplifier circuits, the base current plus the collector current equals the emitter current. The input signal causes variations of base current. The variations of base current cause variations in emitter current with the emitter current variations serving as the output.

7. A - IN THE DTL CIRCUIT OF FIGURE 13, THE VOLTAGE ACROSS THE DIODES IS COMPENSATED FOR BY -- THE VOLTAGE ACROSS THE BASE-EMITTER RESISTANCE OF Q_1 .

When V_{CC} is greater than the input voltage at A, diode D_1 becomes forward biased. This, in turn, develops a voltage across the forward resistance of D_1 . This voltage will be series-aiding with the input voltage, and their sum is the base-to-ground voltage of Q_1 . The voltage at the base of Q_1 together with $-V_{EE}$ forward biases the base-emitter junction of Q_1 . In turn, collector current produces a voltage across the Q_1 emitter junction. The voltage is series opposing the base-to-ground voltage of Q_1 ; therefore, the emitter-to-ground voltage of Q_1 is the difference between the base-to-ground voltage and the voltage across the emitter junction. This voltage is exactly the same as the input voltage. Therefore, the voltage across the diode and the base-emitter junction of Q_1 are opposite and have cancelled each other.

8. B - THE TRANSISTOR AMPLIFIER USED IN THE DTL CIRCUIT OF FIGURE 15 -- IS A COMMON EMITTER TYPE AMPLIFIER.

In a common emitter amplifier, the input signal causes variations of base current. In turn, the base current causes collector current to vary. Collector current variations represent the current output of the transistor.

9. B - IF THE 0 SIGNAL IS A POSITIVE VOLTAGE AND THE 1 SIGNAL IS A NEGATIVE VOLTAGE, THE DTL CIRCUIT OF FIGURE 15 OPERATES AS -- A NOR GATE.

The output at Z will be a logical 0 (a positive voltage) when the inputs at A and/or B are a logical 1. This type of logical decision is characteristic of a NOR gate.

10. A - IN FIGURE 15, SATURATION OF Q_1 IS PREVENTED BY THE CLAMPING NETWORK CONSISTING OF -- D_3 AND $-V_D$.

When the Q_1 collector voltage becomes less than $-V_D$, diode D_3 becomes forward biased and the collector of Q_1 is connected through the forward resistance of D_3 to the negative terminal of $-V_D$. Then, no matter how hard the transistor tries to go into saturation and increase the voltage across R_4 , the collector voltage of Q_1 never decreases more than a diode drop below the value of $-V_D$. The value of $-V_D$ is chosen so it maintains the Q_1 collector voltage just above the saturation point. As a result, Q_1 operates as an unsaturated switch.

QUESTIONS

IMPORTANT—These instructions **MUST** be accurately followed to avoid loss, or errors in grading.

Indicate your answer on this sheet by filling in the box for the most correct answer to each question, as illustrated in the example below.

When all questions have been answered, place the answer card in the proper position to line up the boxes on the card with the boxes on the sheet.

Next, copy the complete lesson code into the space provided on the card, and fill in the answer boxes to correspond with those previously filled in on this sheet.

Before mailing, be certain your correct student number, name and address appear on the card.

LESSON CODE
5215A

Example: Lawns are usually cut with a
 A ☒ (A) lawnmower. (B) hydraulic jack.
 B ☐
 C ☐ (C) razor blade. (D) screwdriver.
 D ☐

1. A ☐ The diode gate circuit of Figure 4 develops a positive output signal, Z, when
 B ☒ (A) both inputs A and B are negative. (B) both inputs A and B are positive. (C) input A is negative and
 C ☐
 D ☐ input B is positive. (D) input A is positive and input B is negative.

2. A ☐ If the 0 signal is +5 volts and the 1 signal is -5 volts, the diode gate of Figure 4 is
 B ☒ (A) an AND gate. (B) an OR gate. (C) a NOR gate. (D) a NAND gate.
 C ☐
 D ☐

3. A ☐ In the diode gate circuit of Figure 6, when input A is POSITIVE and input B is NEGATIVE,
 B ☐ (A) both D_1 and D_2 are cut off. (B) both D_1 and D_2 are conducting. (C) D_1 is conducting and D_2 is cut
 C ☐
 D ☐ off. (D) D_1 is cut off and D_2 is conducting.

4. A ☐ If the 0 signal is +5 volts and the 1 signal is -5 volts, the diode gate of Figure 6 is
 B ☐ (A) an OR gate. (B) a NOR gate. (C) a NOT gate. (D) an AND gate.
 C ☐
 D ☒

5. A ☒ Circuit capacitance C_1 in Figure 15
 B ☐ (A) reduces circuit switching time. (B) prevents instantaneous changes in output Z. (C) increases the
 C ☐
 D ☐ forward resistances of D_1 and D_2 . (D) decreases the forward resistances of D_1 and D_2 .

6. A ☒ The transistor amplifier used in the DTL circuit of Figure 13
 B ☐ (A) is an emitter follower type amplifier. (B) is a common emitter type amplifier. (C) has a high voltage
 C ☐
 D ☐ gain. (D) operates as a saturated switch.

7. A ☒ In the DTL circuit of Figure 13, the voltage across the diodes is compensated for by
 B ☐ (A) the voltage across the base-emitter resistance of Q_1 . (B) V_{EE} . (C) the voltage across R_1 . (D) the
 C ☐
 D ☐ voltage across R_2 .

8. A ☐ The transistor amplifier used in the DTL circuit of Figure 15
 B ☒ (A) has a voltage gain of less than one. (B) is a common emitter type amplifier. (C) does not alter the
 C ☐
 D ☐ logical operation of the diode gate. (D) is operated as a saturated switch to obtain short circuit switch-
 ing time.

9. A ☐ If the 0 signal is a POSITIVE voltage and the 1 signal is a NEGATIVE voltage, the DTL circuit of Figure
 B ☒ 15 operates as
 C ☐
 D ☐ (A) an AND gate. (B) a NOR gate. (C) a NAND gate. (D) an OR gate.

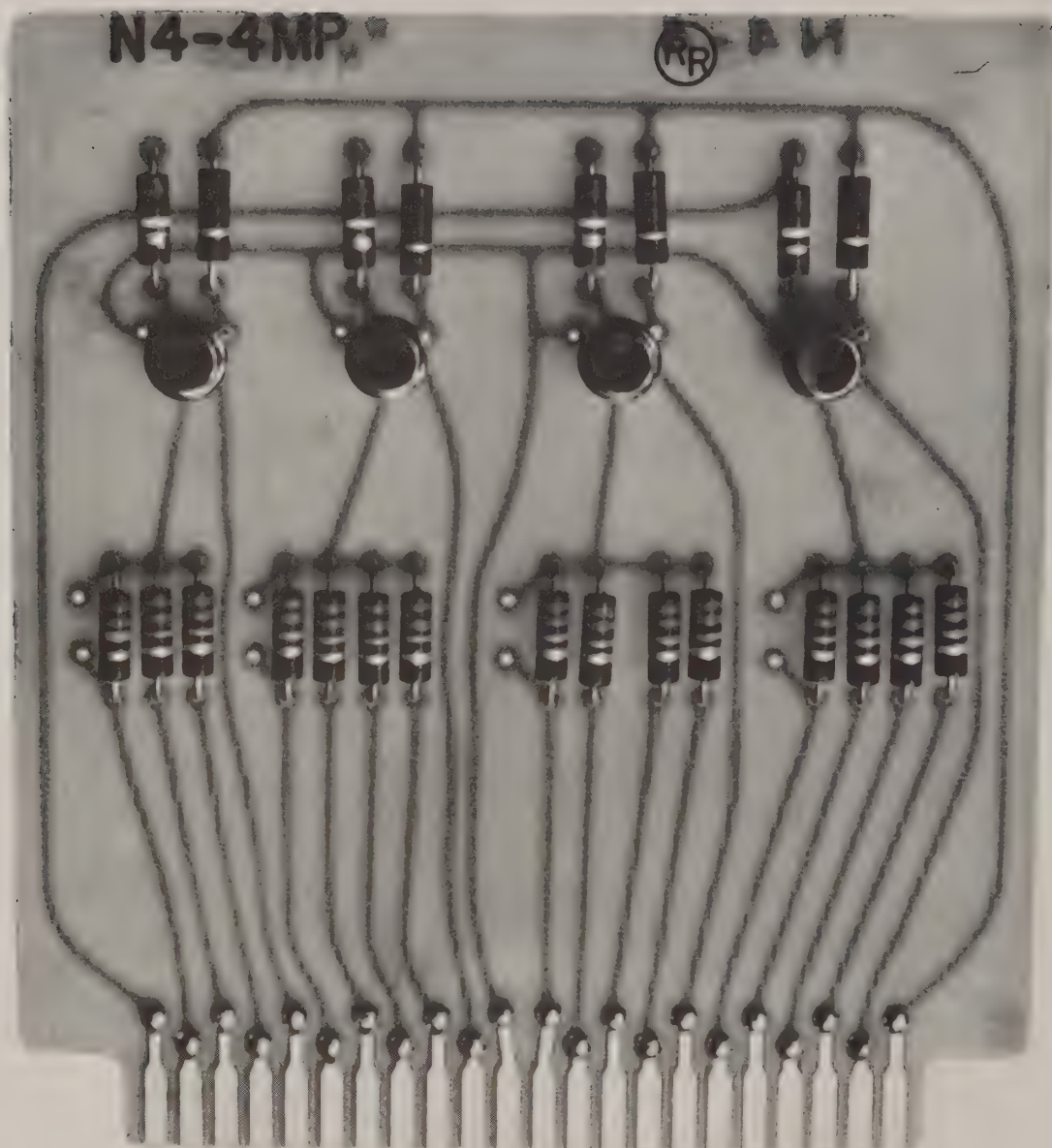
10. A ☐ In Figure 15 saturation of Q_1 is prevented by the clamping network consisting of
 B ☐ (A) D_3 and $-V_D$. (B) R_3 and V_{BB} . (C) C_1 and R_2 . (D) R_4 and V_{CC} .
 C ☐
 D ☐

ACTIVE SWITCHING CIRCUITS

5220

© COPYRIGHT 1973
BELL & HOWELL SCHOOLS, INC.
ALL RIGHTS RESERVED





Plug-in circuit board with four RTL logic gates.

Courtesy Ransom Research, Inc.

CONTENTS

Direct-Coupled Transistor Logic.....	Page 4
Series DCTL Circuits.....	Page 6
Parallel DCTL Circuits.....	Page 9
Diode-Coupled Logic	Page 13
Resistor-Transistor Logic.....	Page 14
Resistor-Capacitor Transistor Logic.....	Page 18
Current-Mode Logic.....	Page 19
Functional Duality.....	Page 25
Appendix—The Basic Truth Tables.....	Page 27

*Perfection is attained by slow degrees; it
requires the hand of time.*

—Voltaire

ACTIVE SWITCHING CIRCUITS

Operation of digital systems is based on the application of specific signal combinations to logical switching circuits. These switching circuits are not restricted to the use of any single type of electronic device. You have learned how diodes may be used in various switching modes, and how they may be combined to form AND and OR gates. You have also learned how diodes and transistors may be combined to form AND, OR, NAND and NOR gates.

Diodes are passive devices. The fact that their internal resistance varies from a low value to an extremely high value, depending on the applied voltage polarity and magnitude, allows them to be used as simple switching devices. However, they can never provide an output signal that is greater than the input signal. After passing through several "stages" of diode switching circuits, signal swing will decrease; and as more gates are added in cascade, the output swing will reduce to an unusable level. It is this disadvantage which brought about the diode-transistor logic combination.

In the diode-transistor logic circuit, the transistor serves more as a "pulse amplifier" than a "switch." However, transistors have characteristics which allow them to be driven in a switching mode. Such transistor circuits are **ACTIVE SWITCHING CIRCUITS** and, therefore, can provide signal amplification (voltage and/or current) along with the switching functions.

In this lesson, we will examine the use of transistors in logical switching circuits in considerable detail. In addition to the diode-transistor arrangement, we will describe **DIRECT-COUPLED TRANSISTOR LOGIC (DCTL)**, **RESISTOR-TRANSISTOR LOGIC (RTL)**, and a special category, known as **CURRENT-MODE LOGIC (CML)**.

DIRECT-COUPLED TRANSISTOR LOGIC

The simplest form of transistor logic is **DIRECT-COUPLED TRANSISTOR LOGIC (DCTL)**. DCTL circuits use a minimum number of parts. The transistors in these circuits act as voltage-controlled switches, very similar in operation to electromechanical relays. The basic type of DCTL circuit is the NOT circuit (or inverter). AND and OR gates can be formed by modifying the NOT circuits. NAND and NOR gates can also be formed by combining the NOT circuits. The truth tables for these functions are presented in an Appendix to this lesson, for easy reference.

A transistor switching circuit connected in the common-emitter configuration develops a polarity reversal between its output voltage and input volt-

age. Figure 1 shows a DCTL transistor NOT circuit. The circuit in Figure 1 is actually a common-emitter transistor amplifier. The input signal at A is applied directly to the base-emitter circuit of Q_1 . The output at Z is taken from the collector of Q_1 to ground. R_1 is the collector load resistor and $-V_{CC}$ is the collector supply. The positive terminal of the collector supply $-V_{CC}$ is connected to ground.

Now let us examine the logical operation of the NOT circuit of Figure 1. For this circuit, assume that the logic 0 level is zero volts and that the logic 1 level is a negative voltage (negative logic). When the input signal at A is zero (logical 0), Q_1 is cut off and the output at Z is highly negative (logical 1). Conversely, when the input signal at A is a negative voltage (logical 1), Q_1 conducts and the output at Z drops to near zero (the logical 0). Summarizing the circuit action, the NOT circuit produces a logical 0 output when a logical 1 input voltage is applied and a logical 1 output when a logical 0 input is applied (shown in Figure 2).

If the logical 0 and 1 voltage levels are reversed to positive logic, the logical operation of the circuit remains the same. When input voltage at A is negative (now logical 0), the output voltage at Z is near zero (now logical 1). Likewise, when the input at A is zero or positive (logical 1), the output voltage at Z is approximately equal to $-V_{CC}$ (the logical 0). This is shown in Figure 2.

The inverter or NOT circuit of Figure 1 is shown using a PNP transistor. If desired, an NPN transistor can be used, then a positive collector supply must also be used. Using an NPN transistor in Figure 1 (with the input at A positive), the emitter junction of Q_1 is forward biased; and Q_1 is driven into saturation. As a result, the output at Z drops to near zero. Conversely, when the input at A is zero (or negative), Q_1 is cut off, and the output at Z rises to approximately the value of the collector supply.

Q_1 in Figure 1 operates as a saturated switch. It is driven into cutoff and saturation. As a result, the output voltage at Z varies from near zero (Q_1 saturated) to approximately the value of the collector supply $-V_{CC}$ (Q_1 cut off). The output voltage at Z has this range only if no load is connected to the output terminals. In an actual digital system, the output terminal of the DCTL NOT circuit would probably be directly connected to the base of a transistor in another DCTL circuit. This connection greatly reduces the output voltage swing of the DCTL gate.

To show how the output voltage is affected, consider the circuit of Figure 3. Here the NOT circuit of Figure 1 is shown connected to another NOT circuit. Assume that the input at A is negative. To saturate Q_1 , the negative input signal at A need only be a few tenths of a volt. The germanium transistors used in DCTL circuits saturate with approximately 0.25 to

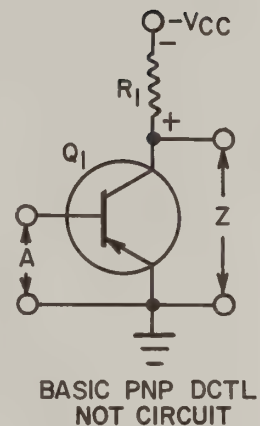


Figure
1

Q_1	A	Z
ON	1	0
OFF	0	1

A
NEGATIVE LOGIC

Q_1	A	Z
ON	0	1
OFF	1	0

B
POSITIVE LOGIC

TRUTH TABLES FOR
DCTL NOT CIRCUIT
OF FIGURE 1

Figure
2

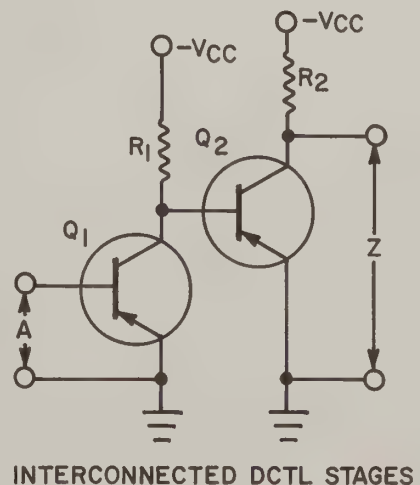


Figure
3

0.4 volt forward bias on their emitter junction ($V_{BE(sat)}$). With Q_1 saturated, its collector current is high and its collector voltage drops to near zero (approximately -0.05 to -0.15 volt for the $V_{CE(sat)}$ of typical DCTL circuit using a PNP germanium transistor). In turn, this voltage is applied to the base-emitter circuit of Q_2 . To drive Q_2 into conduction, the bias on its emitter junction must be -0.25 volt or more negative. However, since Q_1 is saturated, its collector voltage is less than -0.15 volt; therefore, Q_2 is cut off. With Q_2 cut off, its collector current is near zero and the output at Z is approximately equal to $-V_{CC}$.

Now assume that the input at A is decreased to a value less than -0.25 volt. As a result, Q_1 goes into cutoff and its collector-to-ground voltage increases toward $-V_{CC}$. However, as soon as it becomes greater than -0.25 volt, Q_2 starts to conduct. With its emitter junction forward biased, the Q_2 base-to-emitter resistance drops to almost zero. As a result, the collector-emitter circuit of Q_1 is shunted by the low base-emitter resistance of Q_2 . This prevents the collector voltage of Q_1 from becoming more negative than the voltage required to turn on Q_2 , which is usually from -0.25 to -0.4 volt for the transistors used in DCTL circuits. The base-emitter circuit of Q_2 clamps the collector voltage of Q_1 at the emitter-base voltage required to saturate Q_2 . Silicon transistors have a typical $V_{BE(sat)}$ of 0.7 volt and a $V_{CE(sat)}$ of 0.3 volt.

Thus, when a DCTL NOT circuit is connected to another DCTL circuit, the output voltage swing of the first circuit (such as Q_1 in Figure 3) is greatly reduced. Its output voltage varies between the saturated collector voltage of the transistor (0.3 to 0.4 volt for silicon and 0.1 to 0.2 volt for germanium) and the voltage required to saturate the transistor in the next circuit (0.7 to 0.75 volt for silicon and 0.25 to 0.4 volt for germanium). That is, any DCTL circuit which is loaded by another DCTL circuit will have this reduced voltage swing.

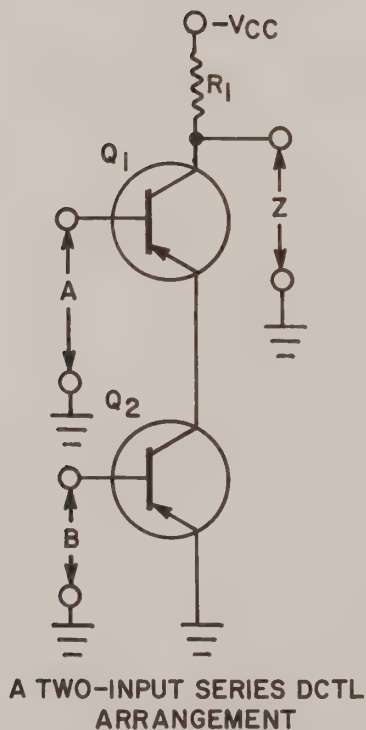


Figure
4

Series DCTL Circuits

The DCTL NOT circuit of Figure 1 can be used to form AND, OR, NAND and NOR gates. NOR and NAND gates may be constructed by interconnecting NOT circuits as shown in Figure 4. Here, two PNP transistors, forming two NOT circuits, are shown connected in series. A single load resistor R_1 and collector supply $-V_{CC}$ serve both transistors. Input signals at A and at B are applied between base and ground of Q_1 and Q_2 , respectively. The output at Z is taken from the collector of Q_1 to the ground, and that output is the sum of the collector-to-emitter voltages of Q_1 and Q_2 .

To examine circuit operation, assume that both inputs at A and at B are zero. Under these conditions both Q_1 and Q_2 are cut off, and the only cur-

rent in R_1 is the small collector cutoff current of both transistors. As a result, the voltages across R_1 is very low; and the collector voltage of Q_1 (output at Z) is approximately equal to $-V_{CC}$.

Now assume that the input at B becomes negative and that the input at A remains at zero. Under these conditions, Q_1 remains cut off and the emitter junction of Q_2 becomes forward biased. However, even though the emitter junction of Q_2 is forward biased, its collector current remains at a low value. Q_1 is still cut off, and it prevents any Q_2 collector current increase. The output at Z is still almost $-V_{CC}$.

The action is similar if the input at A is negative and the input at B is zero. The Q_1 collector current remains at a low value because Q_2 is cut off at this time. Thus, when either or both inputs are at zero, the output at Z is approximately equal to $-V_{CC}$.

Consider the circuit action if the inputs at A and at B are negative enough to saturate both Q_1 and Q_2 . With negative voltages at both input terminals, the emitter junctions of both Q_1 and Q_2 are forward biased; and both Q_1 and Q_2 are driven into saturation. With both transistors saturated, Q_1 and Q_2 collector current increases from its low cutoff value to its high saturated value. As a result, the voltage across collector load resistor R_1 increases; and the collector voltage of Q_1 (output at Z) drops to near zero.

The circuit of Figure 4, like most semiconductor logic circuits, can perform more than one type of logical operation, depending upon the logical 0 and 1 voltage levels chosen. For example, the input signals and the resultant output signals of Figure 4 are shown in the truth table of Figure 5. The plus signs represent near zero or slightly positive voltages, and the negative signs represent voltages sufficiently negative to turn on the transistors. By substituting 0's and 1's into the truth table chart for the + and - signs respectively, the logical operation performed by the circuit can be determined.

For example, assume that the logical 0 voltage level is near zero (or a small positive voltage) and that the logical 1 voltage level is a highly negative voltage. All of the + signs can be replaced by 0's and all of the - signs can be replaced by 1's, as shown in Figure 5. This shows that when both inputs at A and at B are at the logical 1 voltage level (-), the output at Z is at the logical 0 voltage level (+); and when either or both inputs at A and at B are logical 0 (+), the output at Z is logical 1 (-). Under these conditions, the circuit operates as a NOT AND or NAND gate (see Appendix). Since a gate circuit is referred to as a negative type gate if a more negative or less positive voltage is chosen as the logical 1 voltage level, and as a positive type gate if a more positive or less negative voltage is chosen as the logical 1 voltage level, the circuit of Figure 4 is a negative NAND gate.

A	B	Z
+	+	-
+	-	-
-	+	-
-	-	+

A	B	Z
0	0	1
0	1	1
1	0	1
1	1	0

TRUTH TABLE CONSTRUCTION
FOR CIRCUIT IN FIG.4 FOR A
NEGATIVE LOGIC ASSIGNMENT
(NAND)

Figure
5

A	B	Z
+	+	-
+	-	-
-	+	-
-	-	+

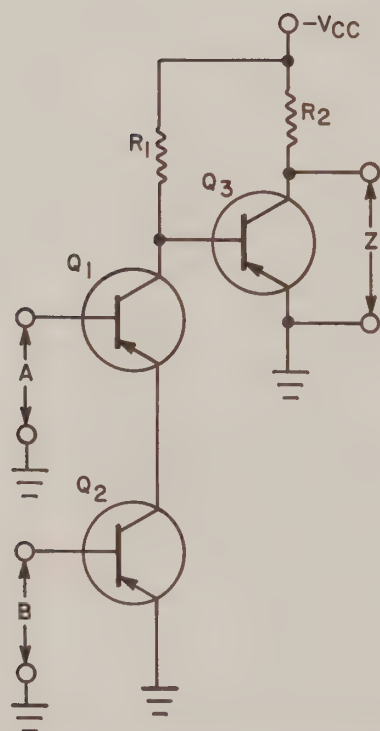
A

A	B	Z
1	1	0
1	0	0
0	1	0
0	0	1

B

TRUTH TABLE CONSTRUCTION
FOR CIRCUIT IN FIG.4 FOR A
POSITIVE LOGIC ASSIGNMENT
(NOR)

Figure 6



LOGIC CIRCUIT OF FIG.4 FOLLOWED
BY A DCTL NOT CIRCUIT

Figure 7

By choosing the 0 and 1 logic levels opposite to those just described, the circuit of Figure 4 functions as a NOR gate. That is, the 0 voltage level would be a highly negative voltage and the 1 voltage level would be near zero (or a slightly positive voltage). Replacing the + signs with 1's and the - signs with 0's (Figure 6) shows that the output for the circuit of Figure 4 is 0 (-), when either or both inputs at A and at B are at the 1 voltage level (+); 1 (+) when both inputs at A and at B are near zero. Under these conditions the circuit operates as a NOT OR or NOR gate. Since the circuit functions as a NOR gate when the 1 voltage level is a zero or positive voltage, it may be called a positive NOR gate.

A	B	Z
+	+	+
+	-	+
-	+	+
-	-	-

A

A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1

B

TRUTH TABLE CONSTRUCTION
FOR CIRCUIT IN FIG.7 WITH
NEGATIVE LOGIC ASSIGNMENT
(AND)

Figure 8

A	B	Z
+	+	+
+	-	+
-	+	+
-	-	-

A

A	B	Z
1	1	1
1	0	1
0	1	1
0	0	0

B

TRUTH TABLE CONSTRUCTION
FOR CIRCUIT IN FIG.7 WITH
POSITIVE LOGIC ASSIGNMENT
(OR)

Figure 9

A negative AND gate or a positive OR gate can be formed by adding a DCTL NOT circuit to the circuit of Figure 4 to invert the output. Such an arrangement is shown in Figure 7. The resulting truth tables are shown in Figures 8 and 9.

Adding the DCTL NOT circuit to the circuit of Figure 4 not only inverts the output signal, but also limits the swing of the output signal of the Q_1Q_2 gate circuit. The output signal swings from near zero volts (when Q_1 and Q_2 are saturated) to the voltage required to drive Q_3 into conduction.

The gate circuits of Figures 4 and 7 have only two inputs. Additional inputs can be added by connecting more transistors in series. However, there is a limit to the number of transistors that may be so connected. As more transistors are connected in series, the voltage from the collector of the top transistor to ground (when all the transistors are in saturation) increases. This voltage is applied to the base of the transistor in the next circuit, such as Q_3 in Figure 7. To keep this transistor (Q_3) cut off, the voltage from the collector of the top transistor to ground must be less than the voltage required to turn on the transistor in the next circuit. For this reason, the number of transistors that may be connected in series is limited.

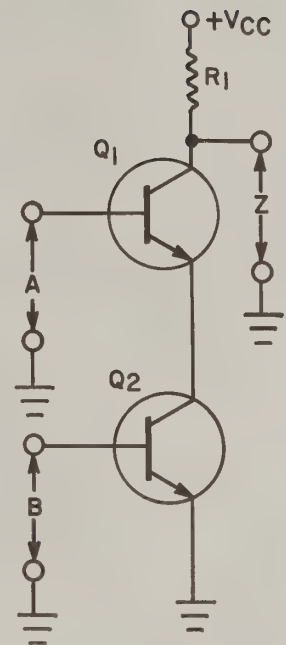
The number of transistors which may be connected in series is also limited by the voltage required to turn on the top transistor. This voltage is higher than that required to turn on the bottom transistor. For example, consider that in Figure 4 both inputs at A and at B are zero. This cuts off both transistors; therefore, their collector currents drop to almost zero. With only leakage current through R_1 , the output at Z increases to nearly $-V_{CC}$. To reduce the output voltage at Z to approximately zero, the base-to-emitter voltage of both transistors must be made sufficiently negative to saturate both of the transistors. This is accomplished for Q_2 by making the input voltage at B slightly negative. However, this is not the case for Q_1 . To saturate Q_1 , the input at A must be more negative than the collector voltage of Q_2 . For example, if a -0.3 volt input signal (typical voltage level for germanium transistors) is required to cause Q_2 to saturate, and the saturation collector voltage of Q_2 is -0.15 volt, then when Q_2 is in saturation, the input signal at A must be $-0.3 - 0.15$ volt, or -0.45 volt to forward bias Q_1 to saturation.

The circuit of Figure 4 uses PNP transistors. If NPN transistors are used, the logical operation performed by the circuit changes. Figure 10 shows a series DCTL circuit employing NPN transistors. When either or both of the inputs at A and at B are at zero or at a slightly negative voltage, either or both transistors are cut off; and the output at Z is approximately equal to V_{CC} . Conversely, when the input voltages at A and B are both positive enough to turn on Q_1 and Q_2 , the output at Z drops to near zero. This circuit functions as a positive NAND gate if the 0 logic level is zero or a negative voltage and if the 1 logic level is a positive voltage. However, if the 0 logic level is zero or a small positive voltage and if the 1 logic level is a negative voltage, the circuit functions as a negative NOR gate.

Parallel DCTL Circuits

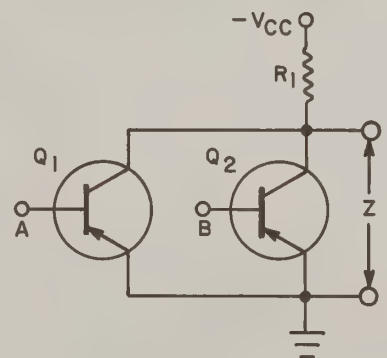
AND, OR, NAND and NOR gates can also be formed by connecting DCTL NOT circuits, such as the one in Figure 1, in parallel. Figure 11 shows such an arrangement. Here, two PNP transistors (two NOT circuits) are connected in parallel to form a gate. A single load resistor R_1 and collector supply $-V_{CC}$ serve both transistors. Input voltages at A and at B are applied between the base and ground of Q_1 and Q_2 , respectively. The output voltage at Z is the collector-to-ground voltage of Q_1 and Q_2 .

To examine circuit operation, assume that the inputs at A and at B are both zero volts. With both inputs at zero volts, the bias on the emitter junction of each transistor is zero; and both Q_1 and Q_2 are cut off. As a result, only the small collector cutoff current of Q_1 and Q_2 flows through R_1 ; and the voltage across R_1 is very small. With a small voltage across R_1 , the collector voltage of both transistors (output at Z) is approximately equal to $-V_{CC}$.



THE TWO-INPUT SERIES DCTL ARRANGEMENT WITH NPN TRANSISTORS

Figure 10



A TWO-INPUT PNP PARALLEL DCTL ARRANGEMENT

Figure 11

Now consider circuit action if the voltage at input A is zero and the input at B is negative enough to saturate Q_2 . Under these conditions Q_1 remains cut off. However, Q_2 is driven into saturation. Q_2 collector current in R_1 reduces Q_2 collector voltage (the output at Z) to its saturated value (near zero).

A similar action occurs if the input at B is zero and the input at A is negative enough to saturate Q_1 . Under these conditions, Q_1 is saturated and Q_2 is cut off. The output at Z is equal to the saturated collector voltage of Q_1 which (like that of Q_2) is near zero.

An output at Z of near zero volts is also obtained when both of the inputs at A and at B are negative enough to saturate both transistors. Under these conditions, both Q_1 and Q_2 are saturated and the output at Z drops to near zero volts.

A	B	Z
+	+	-
+	-	+
-	+	+
-	-	+

A B

TRUTH TABLE CONSTRUCTION
FOR CIRCUIT IN FIG. 11 WITH
NEGATIVE LOGIC ASSIGNMENT
(NOR)

Figure
12

The parallel DCTL gate circuit of Figure 11, just like the series DCTL gate circuit of Figure 4, is either a NAND or NOR gate, depending upon the 0 and 1 logic level polarity chosen. To show the logical operations performed by this circuit, a truth table showing the input and the resultant output signals of the circuit of Figure 11 appears in Figure 12. The + signs represent near zero (or small positive) voltages, and the - signs represent highly negative voltages. By substituting 0's and 1's into the truth table for the + and - signs respectively, the logical operation performed by the circuit can be determined.

A	B	Z
+	+	-
+	-	+
-	+	+
-	-	+

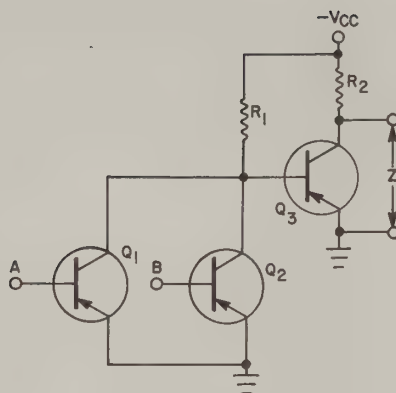
A B

TRUTH TABLE CONSTRUCTION
FOR CIRCUIT IN FIG. 11 WITH
POSITIVE LOGIC ASSIGNMENT
(NAND)

Figure
13

Assume that the 0 logic level is a near zero (or small positive) voltage and the 1 signal level is a highly negative voltage, large enough to drive Q_1 or Q_2 into saturation. Now, in Figure 12A, all of the + signs can be replaced with 0's and all of the - signs with 1's. This conversion shows that the circuit operates as a NOT OR or NOR gate under these conditions (compare Figure 12B to the NOR truth table). Since the circuit of Figure 11 is a NOR gate when the 1 logic level is a negative voltage, the circuit can be called a negative logic NOR gate.

By choosing the 0 and 1 logic levels opposite to those described above, the circuit of Figure 11 functions as a NAND gate. That is, the 0 logic level is a highly negative voltage and the 1 logic level is a near zero (or a slightly positive) voltage. Replacing the + signs with 1's and the - signs with 0's respectively, Figure 13 now shows that, under these conditions, the circuit operates as a NOT AND or NAND gate. Since the circuit of Figure 11 operates as a NAND gate when the 1 logic level is positive and the 0 logic level is negative, it can be called a positive logic NAND gate.



LOGIC CIRCUIT OF FIG. 11
FOLLOWED BY A DCTL NOT
CIRCUIT

Figure 14

If just a positive AND gate or a negative OR gate is desired, another NOT circuit can be connected to the output of the circuit of Figure 11. Such an arrangement is shown in Figure 14. The NOT circuit made up of Q_3 and R_2 inverts the output of the Q_1Q_2 gate, thus forming a positive AND gate or a negative OR gate, depending upon the 0 and 1 logic level polarity chosen (Figures 15 and 16).

The gate circuit of Figure 11 has provisions for only two inputs. Additional inputs can be added by connecting more transistors in parallel. However, there is a limit to the number of transistors that may be connected in parallel. The larger the number of transistors there are, the greater are the demands on the collector supply. Each added transistor requires that the current capabilities of the supply be increased.

When the gate circuit of Figure 11 is connected to another DCTL circuit (as in Figure 14) its output signal swing is greatly reduced. That is, the output swing of Q_1 and Q_2 of Figure 14 is greatly reduced because of the connection of the base of Q_3 to the collectors of Q_1 and Q_2 . The output voltage swing of Q_3 is not reduced and can swing between near zero volts to almost $-V_{CC}$. Q_3 limits the collector voltage of Q_1 and Q_2 to its base-to-emitter turn-on voltage and, therefore, the Q_1Q_2 collector voltage swing is only a few tenths of a volt (typical voltage level for germanium transistors).

If NPN transistors are used in the gate circuit of Figure 11, the polarity of the collector supply must be reversed. In addition, the logic function of the circuit changes. The NPN circuit of Figure 17 is a negative NAND gate and a positive NOR gate.

A	B	Z
+	+	+
+	-	-
-	+	-
-	-	-

A

A	B	Z
0	0	0
0	1	1
1	0	1
1	1	1

B

TRUTH TABLE
CONSTRUCTION FOR CIRCUIT IN
FIG. 14 WITH NEGATIVE
LOGIC ASSIGNMENT
(OR)

Figure
15

A	B	Z
+	+	+
+	-	-
-	+	-
-	-	-

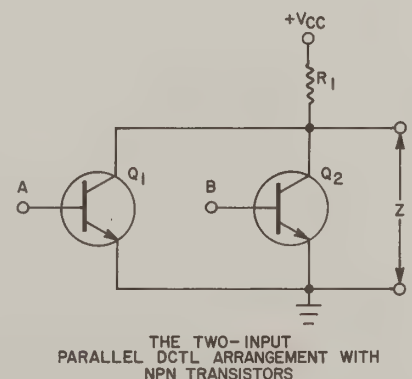
A

A	B	Z
1	1	1
1	0	0
0	1	0
0	0	0

B

TRUTH TABLE
CONSTRUCTION FOR CIRCUIT IN
FIG. 14 WITH POSITIVE
LOGIC ASSIGNMENT
(AND)

Figure
16



THE TWO-INPUT
PARALLEL DCTL ARRANGEMENT WITH
NPN TRANSISTORS

Figure
17

One of the advantages of DCTL circuits is that they require a minimum number of components; one transistor for each input, a single load resistor and a collector supply. No coupling components such as diodes, coupling capacitors, or transformers are needed. However, if a large number of inputs are required, a large number of transistors are also needed; one transistor for each input.

Another advantage of DCTL circuits is that their relatively low output voltage swings allow very little power loss in charging and discharging stray circuit capacitance. This fact, together with the use of high-speed switching transistors, results in shorter switching times. Also, due to the relatively small output signal changes, only a low collector supply voltage is needed. DCTL circuits function well with collector supplies of as low as 3 volts (germanium transistors); whereas other types of logic circuits require collector supplies usually greater than 6 volts (germanium transistors) for discrete circuits. Digital integrated circuits usually operate with supply voltages under 6 volts.

The power dissipation of direct-coupled transistor logic circuits is very low. The transistors in the circuit operate as saturated switches: they are either cut off or saturated. In both these operating regions, transistor power dissipation is lowest.

The major disadvantage of DCTL circuits is their limited operating temperature range. Increases in temperature cause increases in the saturated collector voltage of a transistor. Temperature changes also generally cause decreases in the base-to-emitter voltage required to saturate the silicon transistor. Germanium transistors are affected by temperature changes more than silicon. Leakage (I_{co}) is another parameter affected by temperature changes. Changes in these parameters at high operating temperatures result in improper circuit operation. In addition, the saturated operation of the transistors results in current carrier storage, which produces a longer switching time in comparison to that of an unsaturated circuit. Another disadvantage of DCTL circuits is their susceptibility to noise pulses because of the small voltage swings. Since the DCTL circuits require only low input voltages, noise pulses with amplitudes greater than the input voltage swings can operate DCTL circuits and cause the introduction of errors. Precautions must be taken to reduce noise in DCTL circuits.

The small output voltage swing also is a disadvantage because it limits the driving ability of a DCTL circuit. That is, a DCTL circuit does not produce enough output power to drive a large number of additional DCTL circuits.

The following Practice Exercise questions cover the subjects which you have just studied. They are:

DIRECT-COUPLED TRANSISTOR LOGIC

SERIES DCTL CIRCUITS

PARALLEL DCTL CIRCUITS

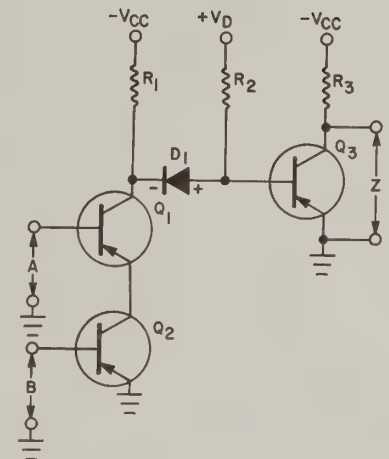
1. The circuit of Figure 1 is (a) an AND gate, (b) a NOT circuit, (c) an OR gate.
2. A NOT circuit is simply an inverting amplifier. True or False?
3. When the NOT circuit of Figure 1 is connected to another DCTL circuit, as in Figure 3, its Q_1 output voltage varies from near zero when Q_1 is saturated to (a) approximately $-V_{CC}$ when Q_1 is cut off, (b) the value of voltage required to drive Q_2 into conduction.
4. Q_1 in the DCTL NOT circuit of Figure 1 operates as (a) a saturated switch, (b) an unsaturated switch.
5. The output at Z of the series DCTL circuit of Figure 4 is equal to the sum of the saturated collector-to-emitter voltages of Q_1 and Q_2 when (a) both of the inputs at A and at B are zero, (b) either input at A or at B is zero, (c) both of the inputs at A and B are highly negative.
6. With a negative logic assignment (1 is negative; 0 is near zero or positive), the circuit of Figure 4 acts as an _____ gate.
7. When the circuit of Figure 4 is operated as a NAND gate followed by a NOT circuit as in Figure 7, the _____ function is produced.
8. The Q_1 collector-to-ground voltage in the DCTL circuit of Figure 7 is approximately equal to $-V_{CC}$ when either or both of the inputs at A and at B are at zero or some slightly positive voltage. True or False?
9. Another input can be added to the circuit of Figure 4 by simply adding another transistor in series with Q_1 and Q_2 . True or False?
10. The output at Z of the parallel DCTL circuit of Figure 11 is approximately equal to the value of $-V_{CC}$ when input signals at (a) A and B are negative enough to saturate both Q_1 and Q_2 , (b) either A or B is negative enough to saturate Q_1 or Q_2 , respectively, (c) both A and B are zero.
11. The output at Z of the parallel DCTL circuit of Figure 14 is near zero when a negative signal large enough to saturate the transistors is applied to both inputs at A and at B. True or False?

12. The DCTL circuit of Figure 11 is a negative OR gate and a positive AND gate. True or False?
13. The addition of a NOT circuit to the circuit of Figure 11 as in Figure 14 results in the entire circuit acting as a positive AND gate and a negative OR gate. True or False?

DIODE-COUPLED LOGIC

Earlier, it was shown that only a limited number of transistors can be placed in series to form a series DCTL circuit. If a number of transistors are stacked in series, the sum of their collector-to-emitter voltages (when they are saturated) may not be low enough to cut off the transistor in the next circuit (especially at elevated temperatures). To overcome this problem, a silicon diode can be used to couple DCTL circuits together. This arrangement produces a **DIODE-COUPLED LOGIC** circuit which is often called a **SILICON-COUPLED TRANSISTOR LOGIC (SCTL)** circuit. Figure 18 shows an SCTL circuit. The addition of silicon diode D_1 allows a greater swing of the collector-to-ground voltage of Q_1 as compared to a conventional DCTL circuit where the collector of Q_1 is directly connected to the base of Q_3 .

The circuit of Figure 18 is similar to the series DCTL circuit of Figure 7 except that a silicon diode D_1 , resistor R_2 , and a positive supply $+V_D$ have been added. Let's consider the series circuit of R_2 , D_1 and R_1 . Notice that R_2 is connected to $+V_D$ and that R_1 is connected to $-V_{CC}$. The two sources (V_D and V_{CC}) are series-aiding and develop a current through the series combination of R_2 , D_1 and R_1 . This current develops a voltage across D_1 of the polarity shown. Assume the voltage across D_1 is 0.4 volt, which is typical for the diodes used in this type of circuit.



USE OF DIODE COUPLING
IN A LOGIC INTERCONNECTION
(AN SCTL CIRCUIT)

Figure
18

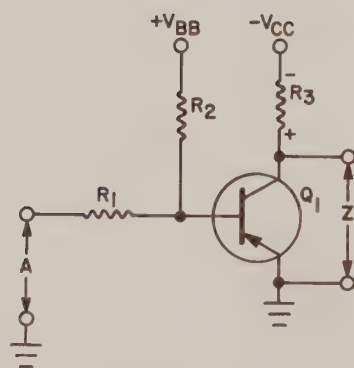
The way the coupling diode affects the circuit operation can be examined by first considering circuit action when the input at A and at B are negative enough to saturate Q_1 and Q_2 . With both Q_1 and Q_2 saturated, the voltage from the collector of Q_1 to ground drops to the sum of the saturated collector-to-emitter voltage of Q_1 and Q_2 . For illustrative purposes, assume that this voltage is -0.3 volt (for germanium transistors). In the circuit of Figure 7, a Q_1 collector-to-ground voltage of -0.3 volt would be large enough to saturate Q_3 and the circuit would not function properly (Q_3 must be cut off when both Q_1 and Q_2 are saturated). The voltage applied to the base-emitter circuit of Q_3 in Figure 18 is not just the Q_1 collector voltage. The voltage applied to the base-emitter circuit of Q_3 is the difference between the voltage across D_1 and the Q_1 collector voltage. With 0.4 volt across D_1 and the Q_1 collector voltage at -0.3 volt, the Q_3 base-to-emitter voltage is $+0.1$ volt. Note: The typical forward drop for silicon diodes is 0.6 or 0.7 volt; however, many silicon diodes can have a drop of as low as 0.4 volt, especially when the current is low for a particular diode. Thus, Q_3 is cut off even though the Q_1 collector voltage is negative enough to drive Q_3 into conduction. The voltage from the collector of the top transistor to ground must be greater than the voltage across the coupling diode before the transistor in the next circuit is driven into conduction. This action permits more transistors to be placed in series; thus, a greater number of inputs can be obtained.

Circuit operation is also affected by the voltage across D_1 when Q_1 , Q_2 , or both are cut off. Under these conditions the Q_1 collector-to-ground voltage rises toward $-V_{CC}$. When the collector-to-ground voltage of Q_1 becomes greater than the voltage across D_1 by the amount necessary to switch on Q_3 , Q_3 saturates; and the clamping action of the Q_3 base-emitter circuit prevents any further change in the Q_1 collector voltage. The Q_1 collector-to-ground voltage rises to the sum of the voltage across D_1 plus the voltage required to turn on Q_3 . For a typical circuit this would be close to -0.7 volt. Therefore, the Q_1 collector voltage in Figure 18 swings from a value equal to the sum of the collector-to-emitter voltages of saturated transistors Q_1 and Q_2 , to a value equal to the Q_3 turn-on voltage plus the voltage across D_1 . In many circuits, this voltage swing amounts to about 0.4 volt, whereas in DCTL circuits, which do not use diode coupling, the collector voltage swing is about 0.2 volt or a little less.

The switching time of Q_3 in Figure 18 is much less than that of Q_3 in Figure 7. The reverse bias provided by V_D and R_2 reduces the time required to remove any current carriers stored in the base region of Q_3 when it is saturated.

RESISTOR-TRANSISTOR LOGIC

In the DCTL circuits described earlier, a separate transistor is required for each input. As a result, the cost of a DCTL circuit increases rapidly as the number of inputs is increased. When a large number of inputs is required with low cost and simplicity, RESISTOR TRANSISTOR LOGIC (RTL) gates can be used. An RTL gate uses only one transistor. Each new input is obtained by adding another input resistor. In addition, the output voltage swing of an RTL circuit is much greater than that of a DCTL circuit. This permits an RTL circuit to drive many more gate circuits than can a similar DCTL circuit.



THE BASIC RTL
NOT CIRCUIT

Figure
19

Figure 19 shows an RTL NOT circuit. The RTL NOT circuit is simply a common-emitter type amplifier. The input signal at A is applied to the base-emitter circuit of Q_1 through R_1 . The output signal at Z is taken from the collector of Q_1 to ground. The signal inversion produced by Q_1 allows the circuit to operate as a NOT circuit. Although the RTL NOT circuit of Figure 19 is somewhat similar to the DCTL NOT circuit of Figure 1, there are two important differences.

One difference is the base resistor R_1 in Figure 19. R_1 limits Q_1 base current. A larger input voltage swing is needed at the input terminals. Only a small portion of this voltage swing reaches the base-emitter circuit of Q_1 , due to the voltage divider action of R_1 and the base-emitter resistance of Q_1 . In addition, R_1 provides some isolation from any previous gate circuits to which the base-emitter circuit of Q_1 might be connected.

The other difference is the reverse bias for the emitter junction of Q_1 , provided by R_2 and $+V_{BB}$. This network keeps Q_1 cut off until the input voltage becomes sufficiently negative to overcome the reverse bias set up by R_2 and $+V_{BB}$.

To examine circuit operation, assume that the input at A is negative enough to overcome the reverse bias set up by R_2 and $+V_{BB}$. As a result, Q_1 is driven into saturation, and Q_1 collector current develops a voltage across R_3 with the polarity shown. In turn, the Q_1 collector voltage (output at Z) drops to near zero (typical voltage level for germanium transistors). As explained earlier, the saturated collector voltage ($V_{CE(sat)}$) is usually a few tenths of a volt.

Conversely, when the input at A is zero (or positive), it aids the reverse bias on the emitter junction of Q_1 . As a result, Q_1 is driven further into cut-off. In turn, collector current decreases to its low cutoff value, the voltage across R_3 decreases to near zero, and the Q_1 collector voltage (output at Z) increases to approximately the value of $-V_{CC}$.

To understand the logical operation of the circuit of Figure 19, assume that logical 0 is zero volts and logical 1 is a highly negative voltage—large enough to saturate Q_1 . When the input at A is a logical 0, the output at Z is close to $-V_{CC}$ or logical 1 (Q_1 is off). Then, when the input at A is a logical 1, the output at Z is near zero volts or logical 0 (Q_1 is saturated). The logical operation is the same if the logical 0 and 1 signal levels are reversed. An NPN transistor could be substituted for Q_1 in Figure 19 providing the polarity of the batteries is reversed.

Figure 20 shows two RTL NOT circuits connected together. To examine the effect of the reverse bias supply and the base resistor on the output voltage swing of the Q_1 stage, first assume that the input at A is zero or positive. Under these conditions, Q_1 is cut off and its collector voltage rises to $-V_{CC}$. The collector voltage of Q_1 is applied to the base of Q_2 through base resistor R_4 . The highly negative collector voltage of Q_1 overcomes the reverse bias on Q_2 developed by R_5 and $+V_{BB}$ and drives Q_2 into saturation. R_4 provides isolation between the collector of Q_1 and the base of Q_2 so that the low base-emitter resistance of Q_2 does not clamp the Q_1 collector voltage at the voltage required to turn on Q_2 , as is done in DCTL circuits. Thus, an RTL circuit provides a higher output voltage when it is connected to another RTL circuit and, as a result, an RTL circuit can drive many additional logic circuits.

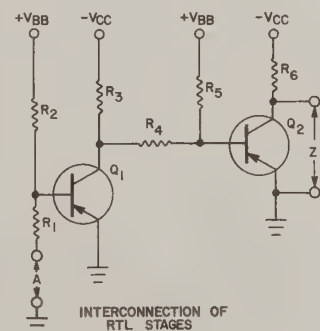
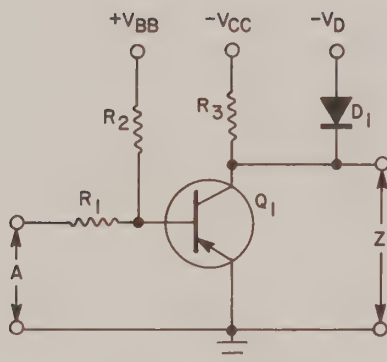


Figure 20

Now assume that the input at A is negative enough to drive Q_1 into saturation. With Q_1 saturated, its collector voltage drops to a few tenths of a volt (typical voltage level for germanium transistors). This small voltage

is not large enough to overcome the reverse bias set up by R_5 and $+V_{BB}$ and, as a result, Q_2 is driven into cutoff. To cut Q_2 off, the forward bias, provided by the Q_1 saturated collector voltage is made less than the reverse bias set up by $+V_{BB}$ and R_5 . Compare this action to that in the DCTL circuits described earlier. In the DCTL circuits, the saturated collector voltage must be a few tenths of a volt to keep the next transistor cut off. This places severe requirements on the transistors used in DCTL circuits. Their saturated collector voltage must be very low to maintain proper circuit operation.

The overall advantages of RTL circuits are increased stability and greater driving ability. The switching speed of an RTL circuit is somewhat higher than that of a DCTL circuit. The reverse bias provided by $+V_{BB}$ and R_5 helps to remove the current carriers stored in Q_2 when it is saturated. As a result, the circuit switching time is small even though saturated operation is employed. Usually, RTL circuits are used in switching systems where the signal frequencies are as high as 20 MHz.



AN RTL NOT CIRCUIT
WITH COLLECTOR CLAMPING

Figure
21

As in DCTL circuits, the transistors in RTL circuits operate as saturated switches. They operate in one of two states: cutoff or saturation. As a result, power dissipation in an RTL circuit is quite small. Losses in RTL circuits are higher than those in DCTL circuits. The voltage swings in an RTL circuit are large. As a result, large currents are needed to charge and discharge stray capacitance.

It is often desired to use a high voltage collector supply in an RTL circuit so high output currents can be obtained. However, large collector voltage swings are undesirable because they waste power in charging stray circuit capacitance. To provide low collector voltage swings with a high collector supply source, an upper level clamping circuit can be used. Figure 21 shows an RTL NOT circuit with upper level clamping provided by D_1 and $-V_D$. D_1 prevents the collector-to-ground voltage of Q_1 from increasing to the value of the collector supply when Q_1 is cut off. The value of $-V_D$ is chosen as the maximum collector voltage desired.

D_1 remains nonconductive as long as the Q_1 collector voltage is less than $-V_D$. This would be the case when Q_1 is saturated and its collector voltage is only a few tenths of a volt (typical voltage level for germanium transistors). However, when Q_1 is cut off, its collector voltage tries to increase to $-V_{CC}$. When the Q_1 collector voltage exceeds $-V_D$, D_1 becomes conductive and the collector of Q_1 is connected, through the low forward resistance of D_1 , to $-V_D$. As a result, the Q_1 collector voltage can never exceed $-V_D$ even though $-V_{CC}$ is much higher than $-V_D$.

The RTL NOT Circuit of Figure 21 may be modified to form either a NAND or a NOR gate. All that is necessary is that additional base resis-

tors be added to provide additional inputs. Figure 22 shows such an arrangement. To examine circuit operation, assume that both inputs at A and at B are zero volts. Under this condition the only bias on Q_1 is the reverse bias set up by $+V_{BB}$; therefore, Q_1 is cut off. With Q_1 cut off, its collector voltage increases toward $-V_{CC}$. However, due to the action of clamping diode D_1 , the Q_1 collector voltage (output at Z) is clamped at $-V_D$. Q_1 would also be cut off if inputs at A and at B were positive, as the positive input voltage would aid the reverse bias developed by V_{BB} and R_3 .

Now assume that either or both the inputs at A and at B are negative enough to overcome the reverse bias set up by $+V_{BB}$ and R_3 . Under these sets of conditions, Q_1 is driven into saturation and its collector voltage (output at Z) decreases to near zero volts.

Depending upon the 0 and 1 logical levels chosen, the circuit of Figure 22 can act either as a NAND gate or a NOR gate. The truth table of Figure 23 shows the input signal conditions and the resultant output signals for the circuit of Figure 22. The + signs represent near zero (or positive) voltages and the - signs represent highly negative voltages.

Notice that this truth table is identical to Figure 12A for the parallel DCTL circuit of Figure 11. Both circuits perform the same logical operations. If logical 0 voltage level is chosen as zero or a positive voltage and logical 1 voltage level as a negative voltage, the circuit of Figure 22 operates as a negative NOR gate.

By reversing the logical 0 and 1 voltage levels, the logical operation performed by the circuit is changed. The circuit functions as a positive NAND gate if the logical 0 voltage level is chosen as a highly negative voltage and the logical 1 voltage level is chosen as zero or a positive voltage.

Only two inputs are shown in the RTL gate circuit of Figure 22. Additional inputs can be added by connecting additional input resistors to the base of Q_1 . However, there is a limit to the number of inputs which may be used. As more inputs are provided, the total base current developed when all the inputs are negative increases. This total base current must be kept below the maximum base current specified for the particular transistor used. Although the values of the input resistors can be increased to reduce base current, each resistor value must be low enough to allow a single negative input signal to turn on the transistor. The considerations mentioned earlier in connection with the RTL NOT gate of Figure 19 also apply to the gate circuit of Figure 22.

If an RTL AND or OR gate is desired, a NOT circuit may be added to the output of the circuit of Figure 22. Figure 24 shows this arrangement.

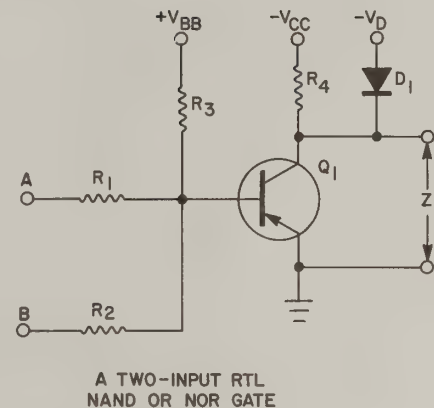


Figure 22

A	B	Z
+	+	-
+	-	+
-	+	+
-	-	+

TRUTH TABLE FOR CIRCUIT IN FIG. 22

Figure 23

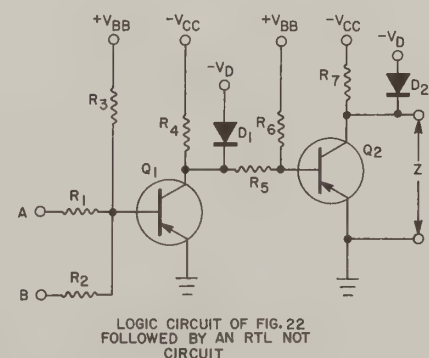


Figure 24

Now the entire circuit functions as a negative OR gate or as a positive AND gate, depending upon the logical 0 and 1 voltage levels chosen. Although PNP transistors are used in Figure 22. NPN transistors may also be used. The polarities of the supply voltages must then all be reversed.

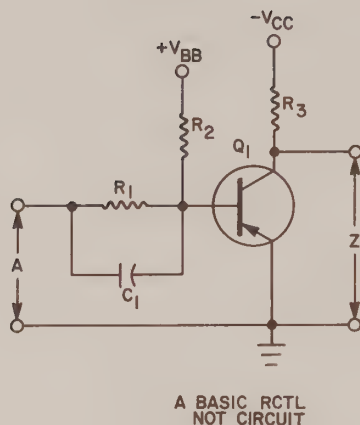


Figure 25

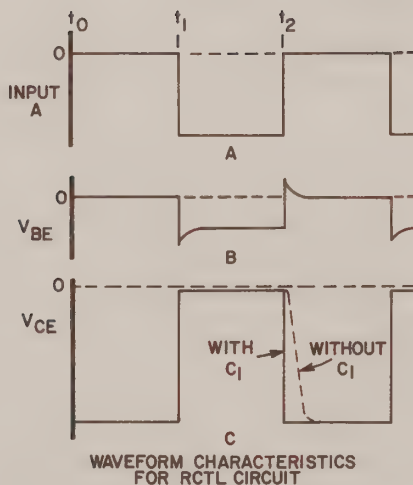


Figure 26

RESISTOR-CAPACITOR TRANSISTOR LOGIC

By placing a capacitor in parallel with the input resistors in an RTL circuit, circuit switching time may be reduced. This type of circuit arrangement is usually referred to as **RESISTOR-CAPACITOR TRANSISTOR LOGIC, (RCTL)**.

Figure 25 shows the RTL NOT circuit of Figure 19 with a capacitor added across the input resistor forming an RCTL circuit. The over-all circuit operation is identical to the circuit operation of Figure 19. When the input at A is negative enough to saturate Q_1 , the Q_1 collector voltage drops to near zero. Conversely, when the input at A is zero or a positive voltage, Q_1 is driven into cutoff and the output at Z becomes approximately equal to $-V_{CC}$.

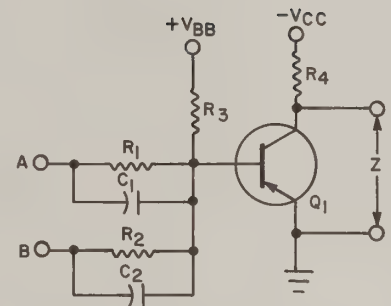
Capacitor C_1 has no effect on the logical operation performed by the circuit. However, it helps to reduce the circuit switching time when the input signal level is changing. To illustrate this effect, assume that the input at A has the waveform shown in Figure 26A. Between t_0 and t_1 , the input at A is zero volts. As a result, the base voltage V_{be} of Q_1 , as shown in Figure 26, is also zero. At t_1 , the input at A changes instantaneously to a highly negative value, which is large enough to saturate Q_1 . This instantaneous change at input A represents a very high frequency. The reactance of C_1 is very low and its shorting effect on R_1 allows all of the input at A to be instantaneously applied to the base of Q_1 . This is only an instantaneous action and as soon as C_1 charges, the base voltage V_{BE} drops to its low normal value. This is the result of the voltage divider action of R_1 and the base-emitter resistance of Q_1 . This action is shown by the pulse at t_1 in the base voltage waveform of Figure 26. The high pulse of base-voltage at t_1 drives Q_1 toward saturation. It actually overdrives Q_1 . This in effect reduces the time required to drive Q_1 from cutoff to saturation and, as a result, reduces circuit switching time.

A similar action also occurs when the input at A changes from its high negative value back to zero volts at t_2 in Figure 26A. When the input at A drops to zero volts, C_1 discharges. It discharges through the base-emitter resistance of Q_1 and the input source. This produces a voltage across the emitter junction of Q_1 which makes the base positive with respect to the emitter. Thus the capacitor aids in the removal of minority carrier charge stored in the base. This aids in the reverse bias set up on Q_1 by $+V_{BB}$ and

drives Q_1 far into cutoff. However, as soon as C_1 discharges, the reverse bias on Q_1 decreases to the value determined by $+V_{BB}$. The over-all effect is that Q_1 is quickly driven from saturation to cutoff by the large pulse of reverse bias and circuit switching time is reduced.

The waveform at the collector of Q_1 in Figure 25 is shown in Figure 26C. The capacitor C_1 has little effect on the collector waveform when the transistor goes from cutoff to saturation. However, C_1 can make a great improvement in the waveshape when the transistor goes into cutoff (at t_2). Figure 26C shows how C_1 reduces the fall time at t_2 . C_1 is usually in the order of 100 pF.

Capacitors may also be added across the input resistors in the RTL gate circuit of Figure 22. Figure 27 shows such an arrangement. The capacitors connected across the input resistors are often called "speed-up" capacitors, as they help to speed up circuit switching. As in the RCTL NOT circuit of Figure 25, the capacitors across the input resistors in Figure 27 do not affect the logical operation of the circuit. The circuit functions in the same way as the RTL circuit of Figure 22.



TWO-INPUT RCTL GATE SIMILAR TO CIRCUIT IN FIG. 22

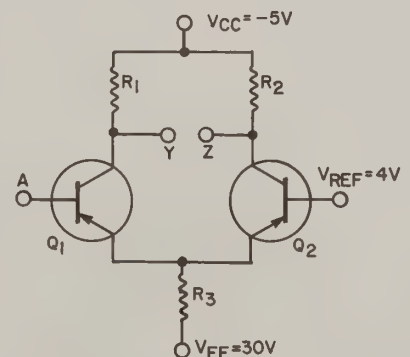
Figure 27

RCTL circuits have the shortest switching times of all the logic circuits that use saturated operation. They provide the low power dissipation obtained with saturated operation, along with short switching times.

X CURRENT-MODE LOGIC

The transistors in DCTL, RTL, and RCTL circuits operate as saturated switches. They operate in their cutoff and saturated operating regions. As a result, circuit switching time is lengthened because of current carrier storage. To eliminate the problem of current carrier storage, unsaturated operation of the transistors is employed. In this form of operation, the transistor may operate between its cutoff and active regions, but is never driven into saturation. One form of logic which employs this unsaturated operation is **CURRENT-MODE LOGIC (CML)**.

Figure 28 shows the basic CML circuit. CML circuits are also called **EMITTER-COUPLED LOGIC (ECL)**. The circuit shown consists of two PNP transistor stages, Q_1 and Q_2 . NPN transistors may also be used if the polarities of the supply and the bias batteries are reversed. Q_1 operates as a phase-splitter, one of whose outputs drives Q_2 , which in turn operates as a common-base amplifier, and the output Y at the collector of Q_1 is the inverting output of the circuit. The output Z of Q_2 is the noninverting output. The input at A is applied between the base of Q_1 and ground. V_{REF} , V_{EE} , and $-V_{CC}$ are the supply voltages for the circuit.



THE BASIC CML CIRCUIT WITH PNP TRANSISTORS

Figure 28

In Figure 28, let us assume two conditions: first, that the input at A is at +3 volts (in reference to ground), and second, that the input is changed to +5 volts. Also, let us assume that the transistors are silicon ($V_{BE} = 0.7$ volt), that R_1 is 1000 ohms, and that each output swings between -3 and -5 volts.

When the input at A is at +3 volts while the reference is +4 volts, Q_1 conducts while Q_2 is cut off because the Q_1 base is less positive than the base of Q_2 . The emitters of the two transistors are at +3.7 volts. Since R_1 is 1000 ohms and the drop across it is 2 volts, the current through Q_1 is then 2 mA. If the beta of the transistors is 100 or more, we can ignore the base current in determining the value of R_3 because the error would be only one percent. The drop across R_3 is $30 - 3.7$ or 26.3 volts, and that divided by 2 mA gives 13,150 ohms. No current flows through Q_2 since it is cut off. The drop across the entire circuit is 35 volts; then the V_{CE} of Q_1 is $35 - 2 - 26.3$ or 6.7 volts. Q_1 is definitely in the active region. The output voltages are -3 volts at Y and -5 volts at Z.

However, when the signal at A is changed to +5 volts, Q_1 cuts off, and Q_2 conducts. This time the base of Q_2 is less positive, causing its emitter junction to be forward biased while the Q_1 emitter junction is reverse biased. The emitters are now at +4.7 volts, and the drop across R_3 is 25.3 volts. Then the current through R_3 is 25.3 volts divided by 13,150 ohms, giving 1.92 mA. Q_2 is conducting with its collector at -3 volts. R_2 has a drop of 2 volts, and its resistance is 2 volts divided by 1.92 mA or 1040 ohms. The V_{CE} of Q_2 is $35 - 2 - 25.3$ or 7.7 volts; Q_2 is now in the active region instead of Q_1 . Therefore, when input A is at +5 volts, Q_1 is driven toward cutoff, Q_2 conducts, the output at Y is -5 volts and the output at Z is -3 volts.

Note that the total supply voltage is 35 volts but that the output voltage swing is only 2 volts. The output voltage swing is small because R_3 is much larger than R_1 or R_2 . Making R_3 even larger will reduce the output swing even more.

A	Y	Z
+3	-3	-5
+5	-5	-3

A

A	Y	Z
1	0	1
0	1	0

B

TRUTH TABLE CONSTRUCTION FOR CIRCUIT IN FIG. 28

Figure
29

Note that, when the input at A is +3 volts, Q_1 conducts collector current; and when the input at A is +5 volts, Q_2 conducts collector current. The input signal switches circuit current from one transistor to the other. This is the reason for the name current-mode logic, sometimes called **CURRENT SWITCHING**.

Logically, the circuit of Figure 28 can be considered as a NOT circuit, inverting amplifier, or as just a simple noninverting amplifier. To examine the logical operation of the circuit, the input signal levels and the corresponding output signal levels are shown in Figure 29. Let us, in these

CML circuits, choose the 0 logic levels as the more positive and less negative signals. Then the more negative or less positive signal levels become logic 1 levels. This is **NEGATIVE LOGIC** or **INVERTED LOGIC**. For example, the input signal levels are +3 and +5 volts. The +5 volt signal is considered the more positive input signal and the +3 volt signal is considered the less positive input signal. Conversely, the output signals are -3 and -5 volts. The -3 volt signal is considered the less negative output signal, and the -5 volt signal is the more negative output signal, as shown in Figure 30A. When the signal at the input goes from the logic 0 level to the logic 1 level and it goes down or becomes less positive (Figure 30A), this is negative logic. In the same manner, when either output signal goes from logic 0 to logic 1 and it also goes down or becomes more negative, the logic is again negative. Thus, for negative logic, the logic 1 levels are below the logic 0 levels in figures such as Figure 30A; this rule applies whether the logic levels are both positive, both negative, or one is positive (logic 0) and the other negative (logic 1). Using this convention of most positive and most negative signals, the logical operation of the circuit can be determined.

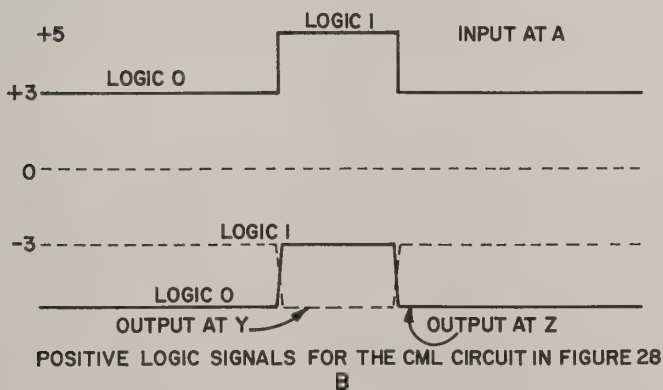
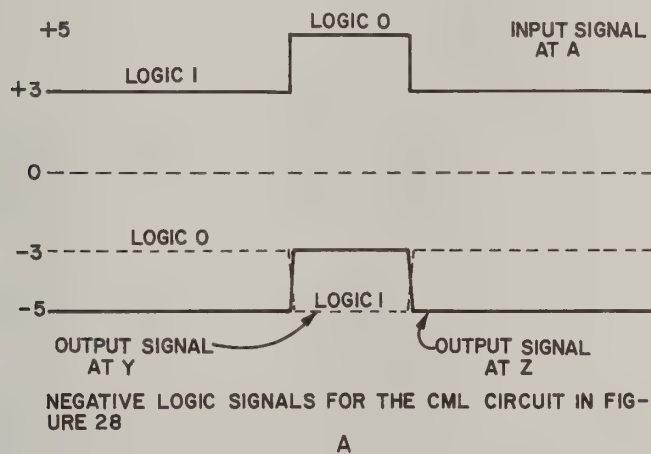
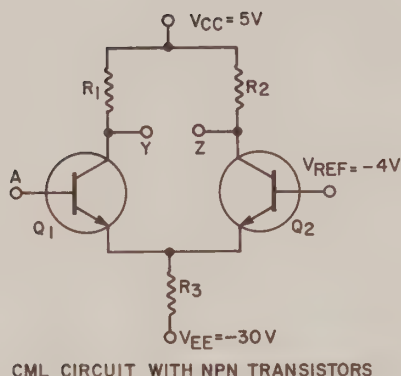


Figure 30

For example, let us assume that the 0 logic level is chosen as the more positive or less negative voltage and that the 1 logic level is chosen as the more negative and less positive voltage. In the truth table of Figure 29, +5 and -3 would be replaced with 0's and +3 and -5 would be replaced with 1's. This conversion produces the truth table shown in Figure 29. Examining this figure shows that, if the Y output is used, the circuit functions as a NOT circuit or inverter. However, if the Z output is used, the circuit functions as a noninverting amplifier. If the 0 and 1 logic levels are reversed as in Figure 30B, the logical operation remains the same.



CML CIRCUIT WITH NPN TRANSISTORS

Figure
31

A	Y	Z
-5	+5	+3
-3	+3	+5

INPUT AND OUTPUT
SIGNAL LEVELS
FOR CIRCUIT IN FIG. 31

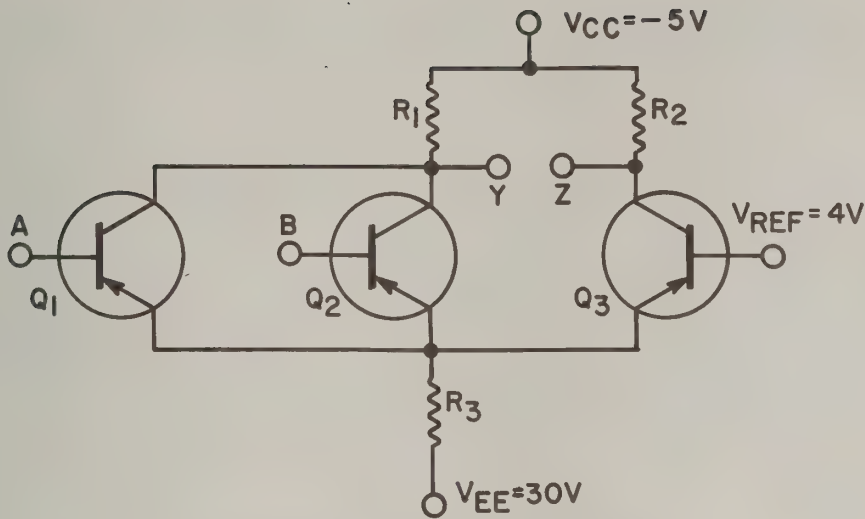
Figure
32

Here we have **POSITIVE LOGIC**. Note that when the input signal goes from logic 0 to logic 1, and it goes up, or more positive (in Figure 30B), this is positive logic because the logic 1 level is above the logic 0 level. When either output goes from logic 0 to logic 1, it also goes up or becomes less negative. Therefore, in positive logic, the logic 1 level is above the logic 0 level whether the two levels are both positive, both negative, or one in positive (logic 1) and the other is negative (logic 0).

PNP transistors are used in the circuit of Figure 28. If NPN transistors are used, the polarities of the bias batteries as well as those of the input signal must be changed. A CML circuit using NPN transistors is shown in Figure 31. Figure 32 shows the input voltage levels and the corresponding output voltages. Circuit operation is the same as that described for Figure 28 except for the difference in signal polarity.

Compare the truth tables for the circuits of Figures 28 and 31. Notice that the input voltage levels used in the PNP circuit of Figure 28 (+3, +5) can be obtained from the output of the NPN circuit of Figure 31. Likewise, the input voltage levels required for the NPN circuit of Figure 31 (-5, -3) may be obtained from the PNP circuit of Figure 28. However, note that the input signals required for the PNP circuit cannot be obtained from another PNP circuit, and likewise for the NPN circuit. As a result, in discrete CML circuits, an NPN current-mode logic circuit is usually driven by a PNP circuit, and vice versa. IC ECL (or CML) circuits do, however, overcome this problem of level shifting.

The basic CML circuits of Figures 28, and 31 can be modified to form AND, OR, NAND, or NOR gates, depending upon the output terminals used and the 0 and 1 logic levels chosen. All that is necessary is that additional inputs be provided by connecting more transistors in parallel with Q₁ in either circuit to provide additional input terminals. Let's use the PNP circuit of Figure 28. Figure 33 shows the circuit with two inputs as well as two outputs and is a **GATING CIRCUIT**, or more simply, **GATE**. The operation of the circuit is the same as that of Figure 28, except that another transistor is provided which can switch circuit current. The input voltages and the resultant output voltages for this circuit are shown in truth



A TWO-INPUT CML CIRCUIT WITH PNP TRANSISTORS

Figure 33

A	B	Y	Z
+3	+3	-3	-5
+5	+3	-3	-5
+3	+5	-3	-5
+5	+5	-5	-3

A	B	Y	Z
0	0	1	0
1	0	1	0
0	1	1	0
1	1	0	1

A
TRUTH TABLE CONSTRUCTION
FOR BOTH OUTPUTS OF FIG.33
WITH POSITIVE LOGIC ASSIGNMENT

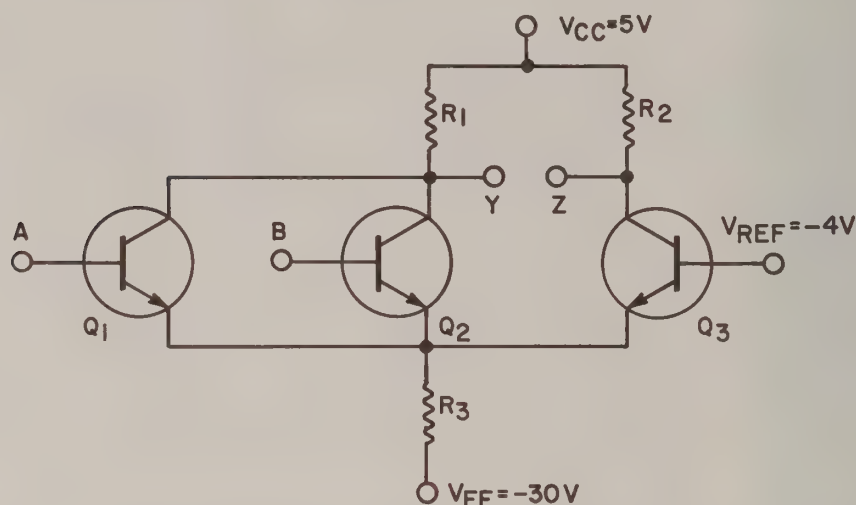
Figure
34

table form in Figure 34. When either or both inputs at A and at B are +3 volts, either or both Q_1 or Q_2 conduct. As a result, the output at Y decreases to approximately -3 volts. In addition, emitter current through R_3 reduces the emitter-to-ground voltage of each transistor. This cuts off Q_3 , as explained earlier. With Q_3 cut off, its collector voltage (output at Z) is approximately the value of V_{CC} or -5 volts.

Conversely, when both the inputs at A and at B are +5 volts, both Q_1 and Q_2 are cut off. In turn, the voltage across R_1 is small and the collector voltage of Q_1 and Q_2 (output at Y) rises to approximately the value of V_{CC} , -5 volts. With Q_1 and Q_2 cut off, the voltage across R_3 would be very small. In turn, the emitter of Q_3 is driven more positive with only +4 volts on its base, as provided by V_{REF} , and Q_3 is driven into conduction, and the voltage across R_3 is about the same as before. As a result, the Q_3 collector voltage (output at Z) changes from -5 to about -3 volts.

To examine the logic operation of the circuit, the 0 and 1 voltage levels must be chosen. Assume that the more positive and less negative voltages, +5 volts on the input and -3 volts on the output, are chosen as the logic 1 levels, and the more negative and less positive signals, +3 volts on the input and -5 volts on the output, are the 0 logic levels. In the truth table of Figure 34A, -3 volts and +5 volts are replaced by 1's, and -5 volts and +3 volts are replaced by 0's. This conversion produces Figure 34B which shows that, if the Y output is used, the circuit is a positive NAND gate. However, if the Z output is used, the circuit is a positive AND gate. If the

logic levels are reversed, the circuit functions as a negative NOR gate if the Y output is used and as a negative OR gate if the Z output is used.



TWO-INPUT NPN CML CIRCUIT

Figure 35

A	B	Y	Z
-3	-3	+3	+5
-5	-3	+3	+5
-3	-5	+3	+5
-5	-5	+5	+3

INPUT AND OUTPUT
SIGNAL LEVELS FOR
CIRCUIT IN FIG.35

Figure
36

Although PNP transistors are used in the circuit of Figure 33, NPN transistors can also be used if the polarity of the bias batteries and the input voltage polarity are reversed. Figure 35 shows the NPN circuit and Figure 36 shows the input voltage levels and the resultant output voltage levels. Although the electrical action of Figure 35 is the same as Figure 31, the logical operation of both circuits is different. For example, assume the more positive signal is the 1 logic level and the more negative signal is the 0 logic level. Then, if the Y output is used, the circuit is a positive NOR gate; and if the Z output is used, the circuit is a positive OR gate. If the signal levels representing logical 0 and 1 are reversed, the operation reverses; the circuit now is a negative NAND gate if the Y output is used, and a negative AND gate if the Z output is used.

There are many advantages of current-mode logic. Due to the unsaturated operation of the transistors, circuit switching time is short. In fact CML (or ECL) switching circuits possess the fastest switching; the rise time is in the order of 10 nsec. Current carrier storage and its effects are negligible. CML circuits are also very versatile. The two output terminals for each circuit permit the circuit to function not only as a NAND or NOR gate, but also as an AND or OR gate. In addition, the output voltage levels of the switching circuits only change a few volts. Thus, with the signal voltage

swings small, large amounts of energy are not wasted in charging circuit capacitance.

One of the disadvantages of CML circuits, as far as discrete circuits are concerned, is that a separate transistor is required for each input, and circuits with a large number of inputs are very expensive. Also, the unsaturated operation of the transistors results in high-power dissipation for the transistors. In respect to problems with noise, there are two situations. One is that internally generated noise is very small because of the differential amplifiers in ECL. The other is external noise, which does present problems because the voltage swing is often a volt or less. In spite of these disadvantages, CML circuits are popular whenever very short switching times are required.

FUNCTIONAL DUALITY

An NPN circuit utilizing positive logic assignments and an identical PNP circuit utilizing negative logic assignments will always perform identical logic functions. However, if negative logic assignments are used with a particular NPN arrangement, the circuit function will not be the same as that associated with positive logic assignments. Similarly, if positive logic assignments are used with a particular PNP arrangement, the circuit function will not be the same as the function associated with negative logic assignments. In fact, the new functions, obtained by inverting the logic assignments, are the DUALS of the original functions. That is, changing the logic polarity assignment of an OR circuit produces the AND function; of an AND circuit, the OR function; of a NOR circuit, the NAND function; of the NAND circuit, the NOR function. You have seen many examples of this principle of DUALITY in this lesson.

The principle of duality does not only apply to transistor logic circuits—it applies to all logic circuits. The principle can be illustrated with any truth table. Figure 37 provides such an example. The truth table for a three-input OR function (regardless of the circuit type) is shown in Figure 37A. We change the logic assignment by converting all 0's to 1's and all 1's to 0's. The result (a three-input AND function) is shown in Figure 37B.

A	B	C	Z
1	1	1	1
1	1	0	1
1	0	1	1
1	0	0	1
0	1	0	1
0	0	1	1
0	1	1	1
0	0	0	0

OR
A

A	B	C	Z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	1	0
1	1	0	0
1	0	0	0
1	1	1	1

AND
B

EXAMPLE OF DUALITY USING A THREE-INPUT TRUTH TABLE

Figure
37

SUMMARY

The simplest forms of logic circuits are direct-coupled transistor logic (DCTL) circuits. The basic DCTL circuit is the NOT circuit. NAND and NOR gates are formed by connecting additional transistors, one for each

input, in series or parallel with the original transistor in the NOT circuit. AND and OR gates are formed by connecting a NOT circuit to the output of the NAND or NOR gate.

The transistors in DCTL circuits operate as saturated switches; therefore, circuit power dissipation is low. However, due to the saturated operation of the transistors, circuit switching time is long in comparison to the switching time of an unsaturated circuit. In addition, the output voltage swing of a DCTL circuit is small. Due to the direct interconnection of circuits, the output voltage varies between the saturated collector-to-emitter voltage of the transistors and the level of voltage required to turn on the next transistor. In most cases this voltage swing amounts to only a few tenths of a volt. This small voltage swing results in an added advantage in that large amounts of energy are not needed to charge circuit capacitance. Sometimes coupling diodes are used to couple DCTL circuits together. This arrangement forms a diode-coupled logic circuit. The coupling diode isolates the stages and permits larger output voltage swings which result in increased driving abilities.

Resistor-transistor logic (RTL) circuits are another form of widely used logic circuitry. The basic RTL circuit is the NOT circuit. By adding additional input resistors, NAND or NOR gates can be formed. If an AND or OR gate is desired, another NOT circuit is added to the NAND or NOR gate. As in DCTL circuits, the transistors in RTL circuits operate as saturated switches with low power dissipation and longer switching times.

The use of resistors as input elements results in large output voltage swings in RTL circuits. This enables an RTL circuit to drive many additional logic circuits. Capacitors placed across the input resistors reduce circuit switching time. This arrangement develops a resistor-capacitor transistor logic (RCTL) circuit.

Very short switching times can be obtained with current-mode logic (CML) circuits. The transistors in these circuits operate as unsaturated switches, resulting in very short switching times and higher power dissipation. The basic CML circuit is a two transistor gate. Both inverted and noninverted outputs can be obtained. By adding additional transistors in parallel with the input transistor, one for each additional input, AND, OR, NAND, or NOR gates can be formed, depending upon the output used and the logical 0 and 1 levels chosen.

Reversing the logic assignments for any given logic circuit will cause the circuit to perform its dual function. The dual of an OR function is the AND function; of the AND function, the OR function; of the NAND function, the NOR function; and of the NOR function, the NAND function.

The following Practice Exercise questions cover the subjects which you have just studied. They are:

DIODE-COUPLED LOGIC

RESISTOR-TRANSISTOR LOGIC

RESISTOR-CAPACITOR TRANSISTOR LOGIC

CURRENT-MODE LOGIC

FUNCTIONAL DUALITY

14. In the circuit of Figure 18 the Q_1 collector-to-ground voltage must be greater than the voltage across D_1 before Q_3 is driven into conduction. True or False?
15. The addition of D_1 in Figure 18 allows more input transistors to be placed in series. True or False?
16. The RTL NOT circuit of Figure 19 is simply a common-emitter type transistor amplifier. True or False?
17. The output voltage swing, collector-to-ground, of Q_1 in the RTL circuit of Figure 20 is (a) larger, (b) smaller than that of the DCTL circuit of Figure 3.
18. In Figure 20, the Q_1 collector-to-ground voltage must overcome the reverse bias set up by $+V_{BB}$ before Q_2 can go into conduction. True or False?
19. What is the purpose of D_1 and $-V_D$ in Figure 21?
20. When both the inputs at A and at B in Figure 22 are zero volts, the output at Z is equal to $-V_D$. True or False?
21. In Figure 22, when the input at A or at B is negative enough to saturate Q_1 , the output at Z is approximately equal to $-V_{CC}$. True or False?
22. What is the function of R_3 and $+V_{BB}$ in Figure 22?
23. Except for the addition of speed-up capacitors, RCTL circuits operate as (a) DCTL circuits, (b) RTL circuits.
24. The capacitors across the input resistors in the gate circuit of Figure 27 help to reduce circuit switching time. True or False?
25. Q_1 in the RCTL gate circuit of Figure 27 operates as (a) a saturated switch, (b) an unsaturated switch.

26. The transistors in a CML circuit operate as (a) saturated switches, (b) unsaturated switches.
27. If the Y output is used in the circuit of Figure 28, the circuit functions as a NOT circuit. True or False?
28. If the Z output is used in the circuit of Figure 33 and the 0 logic level is the more negative and less positive voltage and the 1 logic level is the more positive and less negative voltage, the circuit is (a) an AND gate, (b) an OR gate, (c) a NAND gate, (d) a NOR gate.
29. With positive logic, -1 volt is a logic _____ and -3 volts is a logic _____.
30. When $+2$ volts has a logic 0 and -2 volts has a logic 1, the logic is _____.
31. The logic is _____ when $+2$ volts is a logic 0 while 0 volts is a logic 1.
32. In discrete CML circuits, PNP transistors must be driven by _____ transistors and NPN transistors must be driven by _____ transistors.
33. Given a particular logic circuit, how can you force it to perform its dual function?
34. What is the dual of the AND function?
35. Change all 0's to 1's and 1's to 0's in the NAND truth table provided in the Appendix to this lesson. What function is represented by the resulting truth table?
36. A particular circuit provides the logic $\overline{A \cdot B \cdot C \cdot D} = Z$. Write the expression for the circuit if the logic assignment is reversed.

APPENDIX—THE BASIC TRUTH TABLES

A	B	Z
0	0	0
1	0	0
0	1	0
1	1	1

AND

A	B	Z
1	1	1
1	0	1
0	1	1
0	0	0

OR

A	Z
1	0
0	1

NOT

A	B	Z
0	0	1
1	0	1
0	1	1
1	1	0

NAND

A	B	Z
1	1	0
1	0	0
0	1	0
0	0	1

NOR

IMPORTANT DEFINITIONS

ACTIVE SWITCHING CIRCUITS—Switching circuits which use transistors or multielement vacuum tubes.

CURRENT-MODE LOGIC (CML)—A class of logic circuits in which the logical operations are performed by transistors operating as unsaturated switches. Also known as **EMITTER-COUPLED LOGIC**.

CURRENT SWITCHING—See **CURRENT-MODE LOGIC**.

DIODE-COUPLED LOGIC—A class of logic circuits in which the logical operations are performed by the series or parallel connection of transistors as in **DCTL** circuits. However, the individual logic circuits are connected to other logic circuits through coupling diodes to provide isolation between circuits. Since silicon diodes are widely used as the coupling diodes in these circuits, the circuits are often called silicon-coupled transistor logic circuits and abbreviated **SCTL**.

DIRECT-COUPLED TRANSISTOR LOGIC (DCTL)—A class of logic circuits in which the logical operations are performed by series or parallel-connected transistors. In addition, the logic circuits are directly interconnected. No coupling diodes or resistors are used between individual logic circuits.

EMITTER-COUPLED LOGIC (ECL)—See **CURRENT-MODE LOGIC**.

GATE—Another term for any digital logic circuit having more than one input and one or more outputs and designed in such a way that the output signal or signals depend upon a certain combination of input signals. Also known as a **GATING CIRCUIT**.

GATING CIRCUIT—See **GATE**.

INVERTED LOGIC—See **NEGATIVE LOGIC**.

NEGATIVE LOGIC—Digital logic designed to represent input conditions and output results such that the more negative (or less positive) voltage is a logic 1 while the less negative (or more positive) voltage is a logic 0. In the case where one level is a positive voltage and the other is negative, the positive voltage is a 0 and the negative is a 1. Also known as **INVERTED LOGIC**.

POSITIVE LOGIC—Digital logic designed to represent input conditions and output results such that the more positive (or less negative) voltage is a logic 1 while the less positive (or more negative) voltage is a logic 0. In the case where there is both a positive voltage and a negative voltage, the former is a logic 1 and the latter is a logic 0.

IMPORTANT DEFINITIONS (Continued)

RESISTOR-CAPACITOR TRANSISTOR LOGIC (RCTL)—A class of logic circuit in which the logical operations are performed by input resistors, with a single transistor serving as an inverting amplifier, as in an RTL circuit. In addition, speed-up capacitors are added across the input resistors to improve circuit switching time.

RESISTOR-TRANSISTOR LOGIC (RTL)—A class of logic circuits in which the logical operations are performed by input resistors with a single transistor serving as an inverting amplifier.

SILICON-COUPLED TRANSISTOR LOGIC (SCTL)—See DIODE-COUPLED LOGIC.

PRACTICE EXERCISE SOLUTIONS

1. (b) NOT circuit.—The circuit of Figure 1 is a common-emitter type amplifier. The signal inversion it develops allows it to perform the NOT function.
2. True
3. (b) the value of voltage required to drive Q_2 into conduction.—The low base emitter resistance of Q_2 , when it is forward biased, clamps the collector voltage of Q_1 at the base-emitter voltage required to saturate Q_2 .
4. (a) a saturated switch.— Q_1 operates as a saturated switch since it operates in its cutoff and saturated regions.
5. (c) both of the inputs at A and at B are highly negative.—Since both Q_1 and Q_2 are PNP transistors, they are driven into saturation with highly negative base-to-emitter voltages.
6. NAND
7. AND
8. False—When both of the inputs at A and at B are at zero or some slightly positive voltage, Q_1 and Q_2 are cut off. The Q_1 collector-to-ground voltage tries to increase toward $-V_{CC}$. However, as soon as it become large enough, it drives Q_3 into saturation. The low base-emitter resistance of Q_3 shorts Q_1 and Q_2 , and the Q_1 collector-to-ground voltage is clamped at the value of voltage required to drive Q_3 into saturation.
9. True
10. (c) A and B are zero.—For the output at Z to be equal to $-V_{CC}$, both Q_1 and Q_2 must be cut off. This occurs when both transistors are zero biased, inputs at A and at B at zero volts.
11. False—When both inputs at A and at B are highly negative, both Q_1 and Q_2 are saturated. As a result the voltage applied to the base of Q_3 is very low and Q_3 is cut off. With Q_3 cut off, its collector-to-ground voltage (output at Z) is approximately equal to $-V_{CC}$.
12. False—The circuit of Figure 11 is a negative NOR gate and a positive NAND gate, depending upon the 0 and 1 logic levels chosen.
13. True
14. True

PRACTICE EXERCISE SOLUTIONS (Continued)

15. True
16. True
17. (a) larger—The isolation provided by R_4 in Figure 20 allows the Q_1 collector voltage to swing from its saturated value to approximately the value of $-V_{CC}$.
18. True
19. D_1 and $-V_D$ in Figure 21 act as an upper level clamping circuit. The collector-to-ground voltage of Q_1 cannot exceed the value of V_D (where V_D is a negative supply voltage).
20. True
21. False—Under these conditions Q_1 would be saturated and the output at Z would be the saturated Q_1 collector-to-ground voltage, usually a few tenths of a volt.
22. R_3 and $+V_{BB}$ provide cutoff bias for Q_1 in Figure 22.
23. (b) RTL circuits—An RCTL circuit is formed from an RTL circuit by adding speed-up capacitors across the input resistors.
24. True
25. (a) a saturated switch.— Q_1 operates as a saturated region switch because it operates in its cutoff and saturated regions.
26. (b) unsaturated switches.—The transistors in CML circuits are never driven into saturation. The transistors in CML circuits operate in their cutoff and active regions and thus operate as unsaturated switches.
27. True
28. (a) an AND gate.—When output Z is used, the circuit delivers a logical 1 output (-3 volts) only when both the inputs at A and at B are a logical 1 ($+5$ volts).
29. 1; 0
30. negative
31. negative
32. NPN; PNP

PRACTICE EXERCISE SOLUTIONS (Continued)

- 33. The logic of any circuit can be converted to its dual function by reversing the logic assignment.
- 34. The dual of the AND function is the OR function.
- 35. NOR.
- 36. $\overline{A + B + C + D} = Z$ —The dual of NAND (the original expression) is NOR (the new expression).

1. D - IN DCTL CIRCUITS, SUCH AS THE NOT CIRCUIT OF FIGURE 1, THE TRANSISTORS OPERATE AS -- SATURATED SWITCHES.

Q_1 in Figure 1 operates as a saturated switch. It is driven into cutoff and saturation. As a result, the output voltage at Z varies from near zero (Q_1 saturated) to approximately the value of the collector supply $-V_{CC}$ (Q_1 cut off).

2. A - ASSUMING THAT THE 0 LOGIC LEVEL IS ZERO OR A POSITIVE VOLTAGE AND THE LOGIC LEVEL IS A HIGHLY NEGATIVE VOLTAGE, THE CIRCUIT OF FIGURE 4 ACTS AS A -- NAND GATE. The truth table shown in Figure 5 illustrates that when both inputs at A and B are a highly negative voltage (logical 1), a positive voltage (logical 0) is produced at the output Z. All other combinations of A and B produce a logical 1 at output Z. These conditions describe a negative logic NAND gate. An AND gate would produce a 1 at output Z only when both inputs A and B are 1.

3. D - ASSUMING THAT THE 1 LOGIC LEVEL IS ZERO OR A POSITIVE VOLTAGE AND THAT THE 0 LOGIC LEVEL IS A HIGHLY NEGATIVE VOLTAGE, THE CIRCUIT OF FIGURE 11 ACTS AS -- A NAND GATE.

The truth table shown in Figure 13 illustrates that, when both inputs at A and B are a positive voltage (logical 1), the output at Z is a negative voltage (logical 0). All other combinations of A and B produce logical 1 at output Z. These conditions describe a positive logic NAND gate. An AND gate would produce a 1 at output Z only when both inputs at A and B are 1.

4. A - AN ADVANTAGE OF ADDING A COUPLING DIODE TO A DCTL CIRCUIT AS SHOWN IN FIGURE 18 IS THAT -- MORE INPUT TRANSISTORS CAN BE ADDED.

When a number of transistors are stacked in series, the sum of their collector-to-emitter voltages (when they are saturated) may not be low enough to cut off the transistor in the next circuit. To overcome this problem, a silicon diode can be used to couple DCTL circuits together. The addition of the silicon diode (D_1) allows a greater swing of the collector-to-ground voltage of Q_1 , as compared to a conventional DCTL circuit where the collector of Q_1 is directly connected to the base of Q_3 .

5. D - THE LARGE OUTPUT VOLTAGE SWING OF AN RTL CIRCUIT -- INCREASES THE NUMBER OF ADDITIONAL CIRCUITS IT CAN DRIVE.

When a large number of inputs are required, RTL gates are used. An RTL gate uses only one transistor. Each new input is obtained by adding another input transistor. The voltage swing of an RTL circuit is much greater than that of a DCTL circuit. This permits an RTL circuit to drive many more gate circuits than can a similar DCTL circuit.

6. A - DIODE D_1 AND BIAS SUPPLY $-V_D$ IN THE RTL CIRCUIT OF FIGURE 21 -- LIMIT THE MAXIMUM COLLECTOR VOLTAGE OF Q_1 TO THE VALUE OF $-V_D$.

Figure 21 shows an RTL NOT circuit with upper level clamping provided by D_1 and $-V_D$. D_1 prevents the collector-to-ground voltage of Q_1 from increasing to the value of the collector supply when Q_1 is cut off. The value of $-V_D$ is chosen as the maximum collector voltage desired.

7. A - IN THE RTL CIRCUIT OF FIGURE 22, IF THE 0 LOGIC LEVEL IS A ZERO OR A POSITIVE VOLTAGE, AND THE 1 LOGIC LEVEL IS A HIGHLY NEGATIVE VOLTAGE, THE CIRCUIT ACTS AS -- A NOR GATE.

The truth table shown in Figure 23 illustrates that when inputs at A and/or B are a highly negative voltage (logical 1), the output at Z is a positive voltage (logical 0). When both inputs at A and B are logical 0 (a positive voltage), the output at Z is a logical 1 (a highly negative voltage). These conditions describe a negative logic NOR gate.

8. C - ADDITIONAL INPUTS CAN BE ADDED TO THE RTL CIRCUIT OF FIGURE 22 BY CONNECTING ADDITIONAL -- INPUT RESISTORS TO THE BASE OF Q_1 .

To provide additional inputs to any RTL circuit requires the addition of resistors (one resistor for each additional input) to the base of the input transistor (Q_1 in Figure 22).

9 C- C_1 IN THE RCTL CIRCUIT OF FIGURE 25 -- HELPS TO REDUCE CIRCUIT SWITCHING TIME. C_1 has no effect on the logical operation performed by the circuit. However, it does help to reduce the circuit switching time when the input signal level is changing.

The reactance of C_1 is very low and its shorting effect on R_1 allows all of the input at A to be instantaneously applied to the base of Q_1 . This is only an instantaneous action and as soon as C_1 charges, the base voltage V_{BE} drops to its low normal value. This is the result of the voltage divider action of R_1 and the base-emitter resistance of Q_1 . A high pulse of base-voltage drives Q_1 toward saturation (overdriving Q_1). This, in effect, reduces the time required to drive Q_1 from cutoff to saturation and, as a result, reduces circuit switching time.

10.A - IN CML CIRCUITS, SUCH AS FIGURE 33, -- THE TRANSISTORS OPERATE AS UNSATURATED SWITCHES.

To eliminate the problem of carrier storage, unsaturated operation of the transistors is employed. In this form of operation, the transistor may operate between its cutoff and active regions; however, it is never driven into saturation. CML circuits employ this type of unsaturated operation.

QUESTIONS

IMPORTANT—These instructions **MUST** be accurately followed to avoid loss, or errors in grading.

Indicate your answer on this sheet by filling in the box for the most correct answer to each question, as illustrated in the example below.

When all questions have been answered, place the answer card in the proper position to line up the boxes on the card with the boxes on the sheet.

Next, copy the complete lesson code into the space provided on the card, and fill in the answer boxes to correspond with those previously filled in on this sheet.

Before mailing, be certain your correct student number, name and address appear on the card.

LESSON CODE
5220A

Example:

A	
B	
C	
D	

A wrist watch is a device that (A) indicates time. (B) can be used to trim a lawn. (C) warms water. (D) makes ice.

1.

A	
B	
C	
D	

 In DCTL circuits, such as the NOT circuit of Figure 1, the transistors operate as
(A) unsaturated switches. (B) common-collector type amplifiers. (C) common-base type amplifiers.
(D) saturated switches.

2.

A	
B	
C	
D	

 Assuming that the 0 LOGIC LEVEL is ZERO VOLTS OR a POSITIVE VOLTAGE, and the LOGIC 1 LEVEL is a HIGHLY NEGATIVE VOLTAGE, the circuit of Figure 4 acts as
(A) a NAND gate. (B) an AND gate. (C) an OR gate. (D) a NOR gate.

3.

A	
B	
C	
D	

 Assuming that the 1 LOGIC LEVEL is ZERO OR a POSITIVE VOLTAGE, and that the 0 LOGIC LEVEL is a HIGHLY NEGATIVE VOLTAGE, the circuit of Figure 11 acts as
(A) an OR gate. (B) a NOR gate. (C) an AND gate. (D) a NAND gate.

4.

A	
B	
C	
D	

 An advantage of adding a coupling diode to a DCTL circuit as shown in Figure 18 is that
(A) more input transistors can be added. (B) fewer input transistors can be added. (C) the transistors now operate as saturated switches. (D) the logical operation performed by the circuit is changed.

5.

A	
B	
C	
D	

 The large output voltage swing of an RTL circuit
(A) is a result of the transistor operating as an unsaturated switch. (B) reduces the number of additional circuits it can drive. (C) results in very little energy wasted in charging circuit capacitance. (D) increases the number of additional circuits it can drive.

6.

A	
B	
C	
D	

 Diode D_1 and bias supply $-V_D$ in the RTL circuit of Figure 21
(A) limit the maximum collector voltage of Q_1 to the value of $-V_D$. (B) prevent saturation of Q_1 .
(C) increase the collector voltage swing of Q_1 . (D) keep the emitter junction of Q_1 reverse biased.

7.

A	
B	
C	
D	

 In the RTL circuit of Figure 22, if the 0 LOGIC LEVEL is ZERO VOLTS OR a POSITIVE VOLTAGE, and the 1 LOGIC LEVEL is a HIGHLY NEGATIVE VOLTAGE, the circuit acts as
(A) a NOR gate. (B) an OR gate. (C) an AND gate. (D) a NAND gate.

8.

A	
B	
C	
D	

 Additional inputs can be added to the RTL circuit of Figure 22 by connecting additional
(A) input transistors in parallel with Q_1 . (B) input transistors in series with Q_1 . (C) input resistors to the base of Q_1 . (D) resistors in parallel with R_3 .

9.

A	
B	
C	
D	

 C_1 in the RCTL circuit of Figure 25
(A) helps to increase circuit switching time. (B) reduces the output voltage swing of the circuit. (C) helps to reduce circuit switching time. (D) prevents saturation of Q_1 .

10.

A	
B	
C	
D	

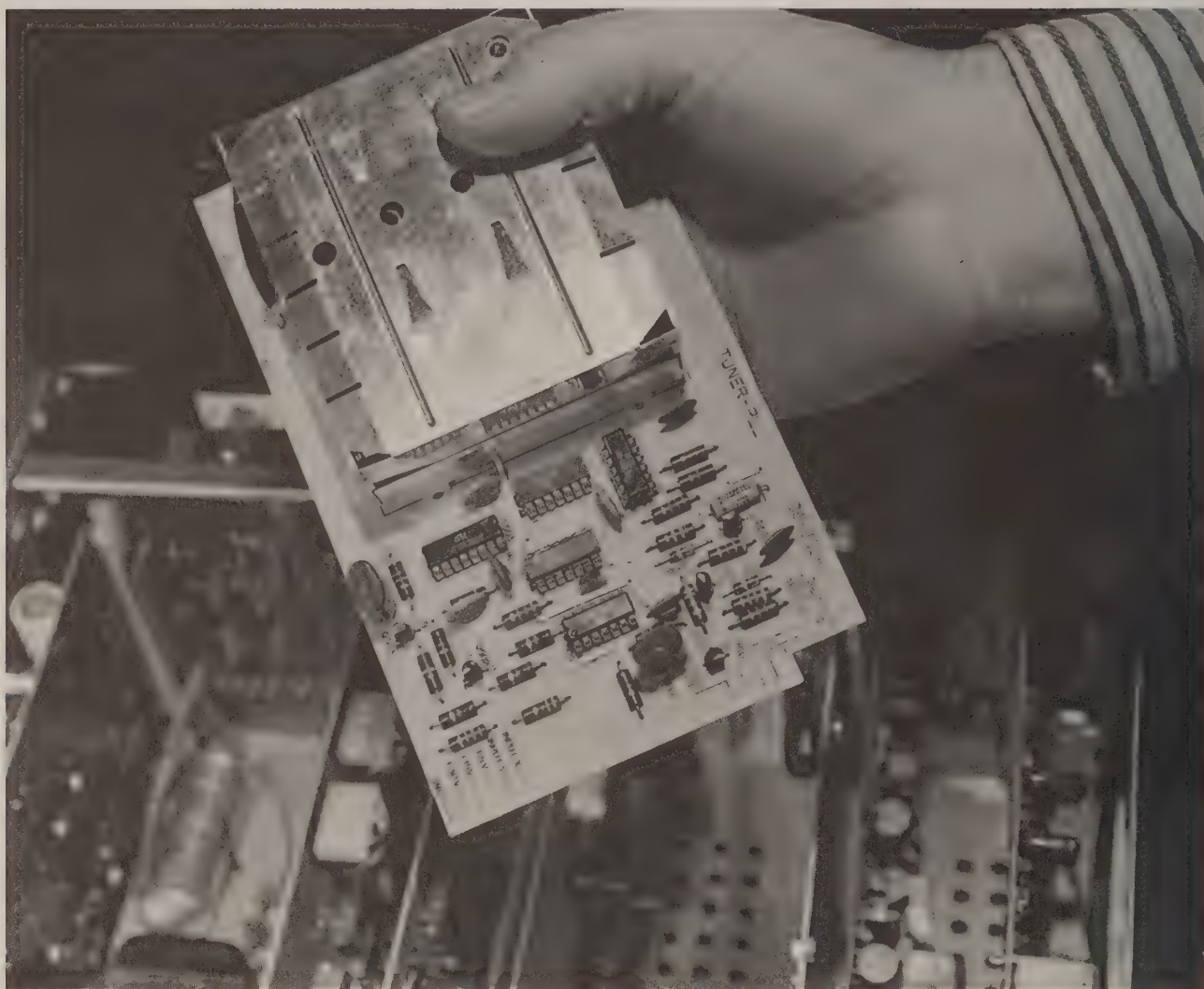
 In CML circuits such as Figure 33,
(A) the transistors operate as unsaturated switches. (B) the transistors operate as saturated switches.
(C) circuit switching time is long. (D) circuit power dissipation is very low.

BASIC COMBINATIONAL LOGIC

5225

© COPYRIGHT 1973
BELL & HOWELL SCHOOLS, INC.
ALL RIGHTS RESERVED





This digital FM stereo tuner uses combinational digital logic in Integrated Circuit form. Shown above is the tuner board which uses NAND gates, OR/NOR gates, and flip-flops.

Courtesy The Heath Company

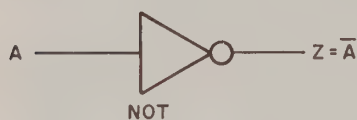
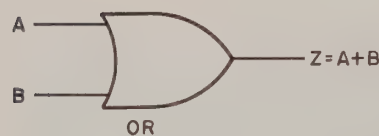
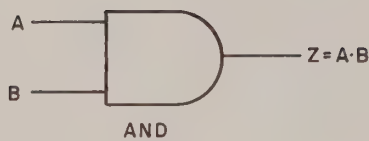
CONTENTS

Timing Diagrams	Page 4
Multilevel Logic Arrangements	Page 8
Multiple-Input Multilevel Logic	Page 10
The Exclusive-OR Function	Page 16
Designing Combinational Logic	Page 22
Switching Functions from Truth Tables	Page 24
Simplification of Switching Functions	Page 27
Implementing the Switching Functions	Page 30
Appendix—Switching Function Relationships and Equivalences ...	Page 37

*Nothing contributes more to
a person's peace of mind
than having no opinions
at all.*

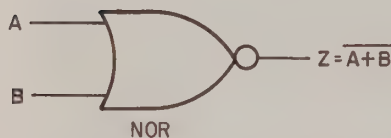
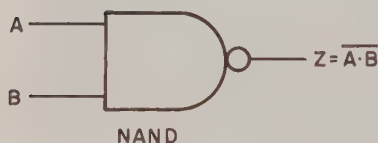
—G. C. Lichtenberg

BASIC COMBINATIONAL LOGIC



THE BASIC FUNCTION SYMBOLS

Figure 1



THE BASIC COMPLEMENTATION SYMBOLS

Figure 2

All applications of digital logic involve a combination of the basic logic functions. These basic functions include AND, OR and NOT, described symbolically in Figure 1. The AND function, also known as **LOGICAL MULTIPLICATION**, and the OR function, also known as **LOGICAL ADDITION**, are dual functions. That is, one performs an operation that is the opposite of the other. The NOT function, also known as complementation, inversion or negation, is used to produce the opposite of a given condition or set of conditions. The complement of the AND function is the NAND function; of the OR function, the NOR function. NAND and NOR are dual functions. AND and OR are also dual functions. The NAND and NOR functions are shown symbolically in Figure 2. These functions are frequently used, in conjunction with NOT, as the basic functions in a digital system.

In this lesson we will examine techniques for analyzing the operation of various combinations of the basic functions. These complex arrangements are known as **COMBINATIONAL LOGIC**. We will also describe how a desired logic arrangement can be designed, starting with a set of logic statements or a truth table. The key to logic analysis and design is the use of Boolean or **SWITCHING ALGEBRA**.

George Boole, an English mathematician and logician (1815–1864), developed a notation system for writing statements of logic in equation form. It is around this system that symbolic logic was developed. For many years, symbolic logic was the only use for the so-called Boolean algebra. However, in recent years it has been discovered to be a powerful tool in the analysis of switching circuits.

With switching functions you can represent a switching circuit in a short, concise equation form. In this form it is easier to visualize the action of the circuit. Also, switching functions may be manipulated mathematically. Each time such manipulation is applied to a switching function, a new function is created which appears different in form; however, mathematically (and logically), it is equivalent to the original. The new switching function found in this way represents a new physical circuit. This new circuit performs the same functions as the original, but has a different physical arrangement of switching elements and (often) a different number of switching elements. Thus, switching algebra may be used to quickly find a number of different switching circuits that perform the same operations.

TIMING DIAGRAMS

The input and output conditions associated with practical digital system applications change from time to time. These changes may be occurring

over either long or very short time periods. Digital systems designed to handle the latter are called DYNAMIC LOGIC systems. They utilize short-duration pulses to represent the various conditions. The electronic waveform representations of such changing conditions can be used to relate digital logic output-input conditions in the same manner as switching functions or truth tables. From a practical viewpoint—especially in locating trouble in a digital system—a **TIMING DIAGRAM** is frequently a more convenient form for showing this relationship.

Figure 3 illustrates typical timing diagrams that may describe the operation of an industrial control unit designed to turn “on” a set of damper motors when A, the temperature reaches 500°F, or B, a ten-minute time interval passes. In Boolean terms:

$$A + B = Z \qquad \text{(OR; INCLUSIVE-OR)} \qquad (1)$$

In Figure 3A, the temperature reaches 500°F (at t_1) before the ten-minute interval passes (at t_2). Between t_1 and t_2 ,

$$A + B = Z = 1 + 0 = 1$$

which is a satisfactory condition for turning “on” the motors at t_1 . Notice that the output signal, Z, rises to the logical 1 value at the instant when A rises to the logical 1 value at t_1 . In a practical system, there are propagation delays associated with the output and input states. These may be very small (nanoseconds) or fairly large (microseconds). In Figure 3B, the ten-minute interval passes (at t_2) before the temperature reaches 500°F (at t_3), where A has a 0, but B has a 1:

$$A + B = Z = 0 + 1 = 1$$

which again is a satisfactory condition for turning “on” the motors. Now the output signal, Z, rises to the logical 1 value at the instant that B rises to the logical 1 value (at t_2). In Figure 3C, both conditions occur simultaneously (at t_2):

$$A + B = Z = 1 + 1 = 1$$

indicating that the **INCLUSIVE-OR** function is implemented. All three signals rise to the logical 1 value simultaneously. In the absence of both input conditions, the output stays at logical 0:

$$A + B = Z = 0 + 0 = 0$$

This can be seen during the time interval before A reaches logical 1 (Figure 3A), before B reaches logical 1 (Figure 3B), and before A and B reach logical 1 (Figure 3C). The truth table conditions which apply to this timing diagram development are presented in Table 1.

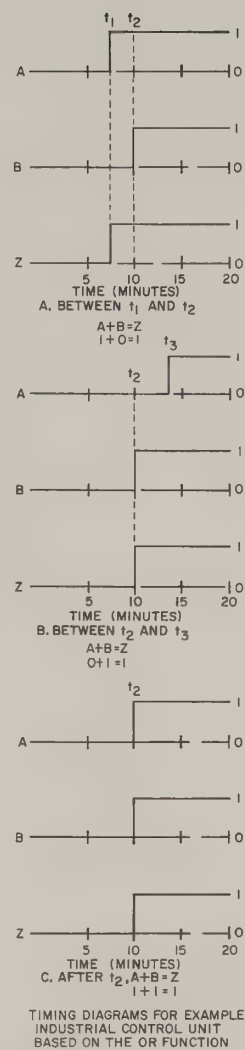


Figure 3

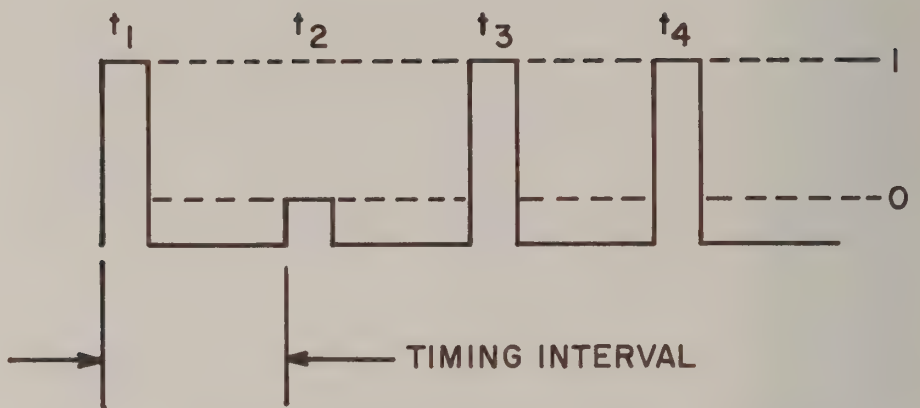
A	B	Z
1	1	1
1	0	1
0	1	1
0	0	0

Truth Data for OR
 $A + B = Z$

Table 1



Figure 5



SEQUENCE OF PULSES REPRESENTING LOGICAL DIGITS

Figure 4

When a digital system uses a series of short-duration pulses for the input and output conditions (as in a digital computer), one can no longer think in terms of a single logical 1 or logical 0 value. Rather, each specific **TIMING INTERVAL** must have a logical 0 or a logical 1 associated with it, indicated by the presence of a logical 0 or a logical 1 pulse, respectively, as in Figure 4. We can call this **DYNAMIC LOGIC**. The time at which the first pulse appears we shall designate t_1 , the second pulse will be at t_2 , the third pulse at t_3 , etc.

Note that Figure 5 is an AND unit with the input at A given as 1100. With positive logic employed, this means that at t_1 , A is a logical 1; at t_2 , A is again a logical 1; and at t_3 and t_4 , A is a logical 0. This can be seen in the top waveform, A, of Figure 6. Here the taller pulses at t_1 and t_2 are logical 1 pulses, and the short pulses at t_3 and t_4 represent logical 0. Similarly, in Figure 5, the input at B is 0101, which then means that at t_1 and t_3 , B is at a logic 0 (short pulses in the middle waveform of Figure 6); while at t_2 and t_4 , B is then at logic 1 (tall pulses). The circuit of Figure 5 is an AND gate (where the output at Z is $A \cdot B$). The truth table governing the operation of this unit appears in Table 2. This means that Z will have a logic 1 pulse only when the input pulses at both A and B are logic 1 pulses. In Figure 6, t_2 is the only time at which both A and B have logic 1 pulses; therefore, at t_1 , t_3 and t_4 , Z has logic 0 pulses. Consequently, $Z = 0100$ as shown for the bottom waveform for Figure 6. The circuit operation can be represented in Boolean terms:

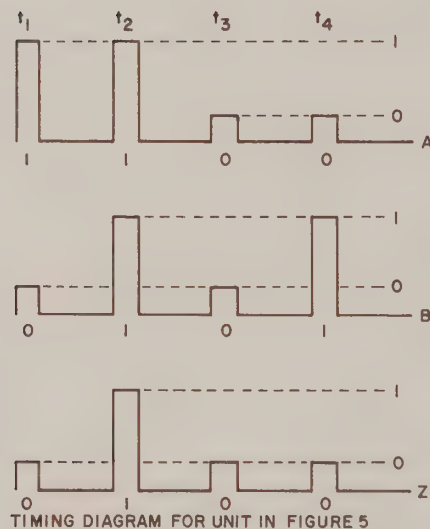


Figure 6

A	B	Z
0	0	0
1	0	0
0	1	0
1	1	1

Truth Data for AND
 $A \cdot B = Z$

Table 2

$$A \cdot B = Z \quad (\text{AND}) \quad (2)$$

In this case,

$$A \cdot B = (1100) \cdot (0101) = 0100 = Z.$$

At t_1 , we can say that $A \cdot \bar{B} = \bar{Z}$ since only A has a logic 1 pulse (B and Z have logic 0 pulses). At t_2 , $A \cdot B = Z$ because both A and B have logic 1 pulses and therefore Z also has a logical 1 pulse. Then at t_3 , $\bar{A} \cdot \bar{B} = \bar{Z}$; and at t_4 , $\bar{A} \cdot B = \bar{Z}$. This represents all possible combinations of inputs.

Notice that parentheses are being used to help distinguish between the series of digits belonging to A and the series belonging to B. This is not essential, but it helps to avoid confusion.

If the two input signals shown in Figure 5 were applied to a NOR unit, implementing the Boolean logic

$$\overline{A + B} = Z \quad (\text{NOR}) \quad (3)$$

the condition shown in Table 3 would govern the construction of the timing diagram. An output pulse will not occur in those intervals when A or B contains a pulse. The timing diagram appears in Figure 7. You should examine this figure and determine the reason for each output interval appearing as it does on the basis of the truth table conditions. The digital equivalent is given by:

$$\overline{A + B} = \overline{(1100) + (0101)} = 0010 = Z.$$

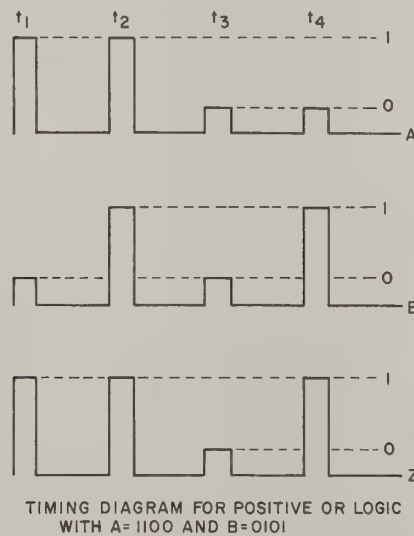


Figure 8

The OR function timing diagram for these input conditions is shown in Figure 8. The OR function truth table, used to support this development, is the same as Table 1. From this:

$$A + B = (1100) + (0101) = 1101 = Z.$$

A	B	Z
1	1	0
1	0	0
0	1	0
0	0	1

Truth Data for NOR
 $\overline{A + B} = Z$

Table
3

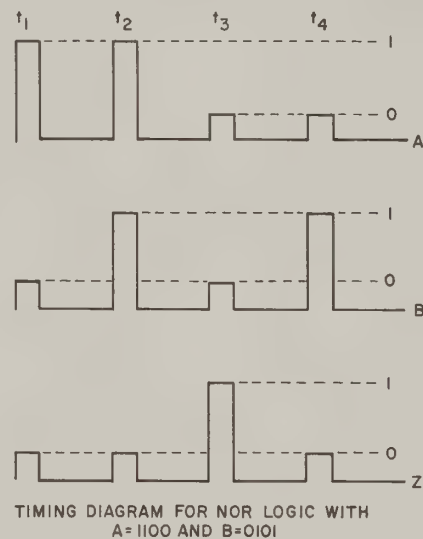
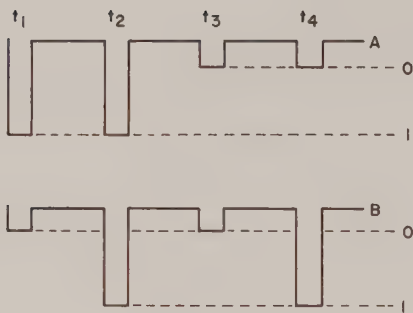
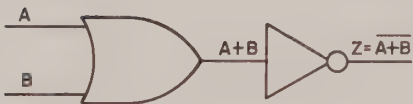


Figure
7



TIMING DIAGRAM FOR NEGATIVE LOGIC
NOR WITH A=1100 AND B=0101

Figure 9



THE OR-NOT MULTILEVEL LOGIC

Figure 10



THE MULTILEVEL NAND-NOT
ARRANGEMENT

Figure 11

A	B	Z
0	0	1
1	0	1
0	1	1
1	1	0

Truth Data for NAND
 $A \cdot B = \overline{Z}$

Table 4

Although all of the digital signals presented thus far have consisted of positive pulses or voltage levels, the same techniques apply to negative polarity signals. However, in order to preserve the same logic operations, the units would have to be based on a negative logic design. Figure 9 illustrates a timing diagram similar to that of Figure 7, but with negative logical signals being used.

MULTILEVEL LOGIC ARRANGEMENTS

The OR-NOT arrangement shown in Figure 10 is a very simple multilevel or combinational logic arrangement used to obtain the NOR function. Such an arrangement is classed as "two-level" logic since two cascaded functions are involved. As shown, the conditions at each level can be determined on the basis of the input conditions and the intervening function. In Figure 10, the input conditions for the first level are A and B. The first level performs the OR function so that the first output is $A + B$. This serves as the input to the second level, which performs the NOT function. With an input of $A + B$, the NOT function output must be $\overline{A + B}$ (the NOR function). Truth tables for multilevel logic can be developed on a level-by-level basis or in terms of the overall function. The latter is preferred since it reduces the work involved considerably, especially for more complex systems. In order to do this, it is necessary to derive and evaluate the overall Boolean expression for the multilevel arrangement.

The DOUBLE NEGATION of any function produces the function itself:

$$\overline{\overline{A}} = A \quad \text{(DOUBLE NEGATION)} \quad (4)$$

Consequently, $\overline{\overline{Z}} = \overline{\overline{A + B}} = A + B = Z$ and $\overline{\overline{(A + B)C}} = (A + B)C$, etc.

Figure 11 presents a multilevel NAND-NOT arrangement. With inputs A and B, the NAND level produces the output $A \cdot B$. With this as an input, the NOT level produces the output $\overline{A \cdot B}$ (read as A AND B double notted). The double negation of the AND function (or the complement of NAND function) produces the basic AND function:

$$\overline{\overline{A \cdot B}} = A \cdot B.$$

Hence, the overall logic arrangement is an AND unit, accomplishing the basic function by utilizing multilevel logic. The truth table for this arrangement is the same as Table 2. The NAND truth table, which governs the operation of the first level in Figure 11, is presented as Table 4. These output conditions serve as possible input conditions to the second level. Since this is an inverting level, all conditions must be complemented. Complementing the output column of Table 4 produces the output column of Table 2 (for the same original input conditions).

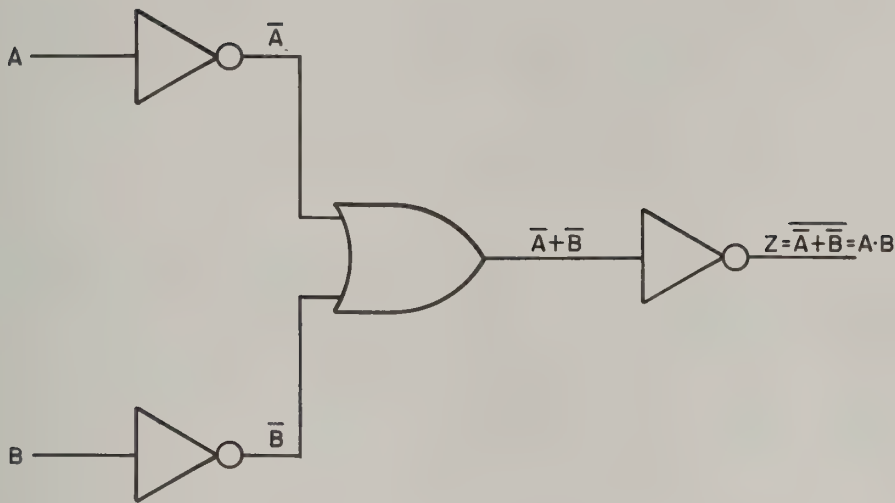
THE THREE-LEVEL $\overline{\overline{A} + \overline{B}}$ LOGIC ARRANGEMENT

Figure 12

Another very simple multilevel logic system is shown in Figure 12. Here, inversion precedes and follows the OR function to produce a three-level ($\overline{\overline{A} + \overline{B}}$) arrangement (read as NOT A OR NOT B notted). With this arrangement, the inputs to the second level are always the inverse of the inputs to the first level. The second-level inputs are then operated on by the OR function whose outputs, in turn, are inverted by the third level. These relationships are shown, in digital form, in Table 5.

First-Level Inputs		Second-Level Inputs		Second-Level Output	Final Output
A	B	$\overline{A}$	$\overline{B}$	$(\overline{A} + \overline{B})$	Z
0	0	1	1	1	0
1	0	0	1	1	0
0	1	1	0	1	0
1	1	0	0	0	1

Level-by-Level Construction of Truth Data for Multilevel Logic in Figure 12

Table 5

“Borrowing” the original input conditions and the final output conditions from Table 5, the truth table for the multilevel logic in Figure 12 can be

A	B	Z
0	0	0
1	0	0
0	1	0
1	1	1

Truth Data for
 $\overline{\overline{A + B}} = A \cdot B = Z$

Table
6

Basic Function	Complement	Dual
$A \cdot B$	$\overline{A \cdot B}$	$A + B$
$A + B$	$\overline{A + B}$	$A \cdot B$
$\overline{A \cdot B}$	$A \cdot B$	$\overline{A + B}$
$\overline{A + B}$	$A + B$	$\overline{A \cdot B}$

Comparison of Basic Complementary and
Dual Functions

Table
7

constructed as shown in Table 6. A comparison of Table 6 and Table 2 shows them to be identical. Hence, the three-level logic in Figure 12 produces the AND function:

$$\overline{\overline{A + B}} = A \cdot B \quad (5)$$

This is to be expected since the principle of dualization states that whenever the logic assignments for a given circuit are inverted, the circuit performs the dual function. Placing the NOT function before and after the OR function in Figure 12 is the same as reversing the normal logic assignments for the OR circuit. Duals are usually (but not always) obtained by redefining the logic assignments rather than by inserting an additional logic function. The remaining basic dual functions are defined by:

$$\overline{\overline{A \cdot B}} = A + B \quad (6)$$

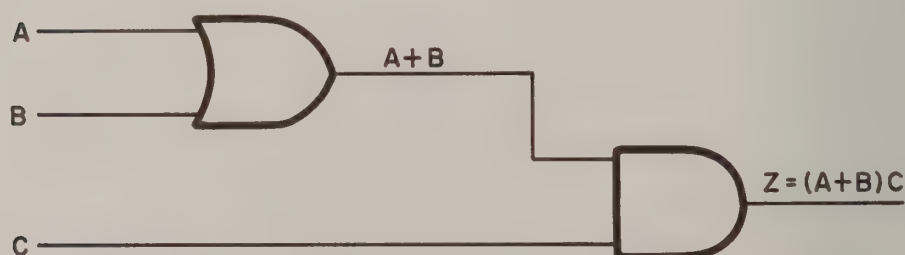
$$\overline{\overline{A + B}} = \overline{A \cdot B} \quad (7)$$

$$\overline{\overline{A \cdot B}} = \overline{A + B} \quad (8)$$

You must be careful not to confuse dualization and complementation. Complementation occurs only when the NOT level follows a function, as described earlier in this section. Complementation does not occur when it precedes a function. A complete comparative summary is presented (for the basic functions) in Table 7.

MULTIPLE-INPUT MULTILEVEL LOGIC

Multilevel logic arrangements may consist of any number of inputs and any number of logic levels. A three-input, two-level arrangement known as OR-AND logic is shown in Figure 13. The inputs for the first level are A and B.



THE OR-AND LOGIC $(A+B) \cdot C$

Figure 13

The function is OR, leading to a first-level output of $A + B$. This output and C form the inputs to the second level. The second-level function is AND, leading to a second-level output of $(A + B) \cdot C$, read "A or B with C anded." The parentheses are used to distinguish between the condition $A + B$ and the condition C . As far as the second level is concerned, $(A + B)$ "behaves" as a single input condition. Table 8 provides the truth table for this arrangement. There are three basic inputs, leading to $2^3 = 8$ possible signal combinations. (There are always 2^n combinations, where n is the number of input conditions.) Since the second level is AND, its output can be logical 1 only when both inputs are logical 1. Hence, whenever C is logical 0, the output must be a logical 0 (regardless of the value of A or B). This allows the construction of the output condition for the second, fourth, fifth and last rows of Table 8. The second-level input $(A + B)$ is a logical 0 only when A and B are both logical 0's. A and B are both 0 in the sixth row of Table 8 and a logical 0 output is assigned to that row. All other combinations of A and B will appear as a logical 1 at the input to the second level, which, in conjunction with a logical 1 at input C , will produce an output logical 1 for the overall arrangement.

A	B	C	Z
1	1	1	1
1	1	0	0
1	0	1	1
1	0	0	0
0	1	0	0
0	0	1	0
0	1	1	1
0	0	0	0

Truth Data for OR-AND
Logic: $(A + B) \cdot C = Z$

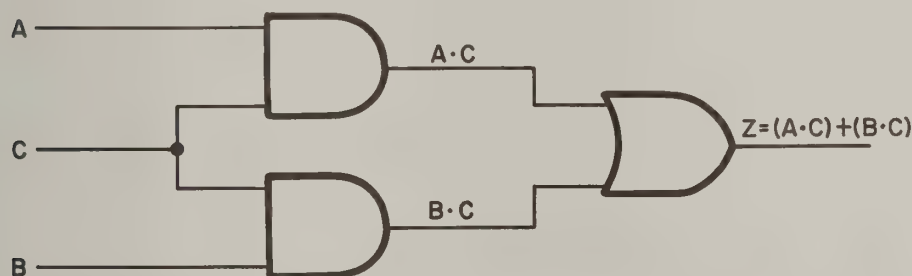
Table
8

The level-by-level construction of the truth table is shown in Table 9. The first-level OR function is identical to any other two-input OR function. This portion of the table is repeated twice—one time for all A and B combinations with C equal to 1, the second time for all A and B combinations with C equal to 0. Since the second level is an AND function, all combinations with C equal to 0 must produce a 0 output (as shown). The logical 1 outputs are formed only when "both" second-level inputs are logical 1's (treating the first-level output as a single condition). The balance of the outputs are logical 0's. The "shortened" truth table is constructed by using the original input columns (A , B and C) and the output column (Z). This, with a rearrangement of rows, provides a summary identical to Table 8.

1st-Level Inputs		2nd-Level Inputs		2nd-Level Outputs
A	B	1st-Level Output ($A + B$)	C	Z
1	1	1	1	1
1	0	1	1	1
0	1	1	1	1
0	0	0	1	0
1	1	1	0	0
1	0	1	0	0
0	1	1	0	0
0	0	0	0	0

Level-by-Level Construction of Truth Data
for $(A + B) \cdot C$ (Figure 13)

Table
9



THE AND-OR LOGIC $(A \cdot C) + (B \cdot C)$

Figure 14

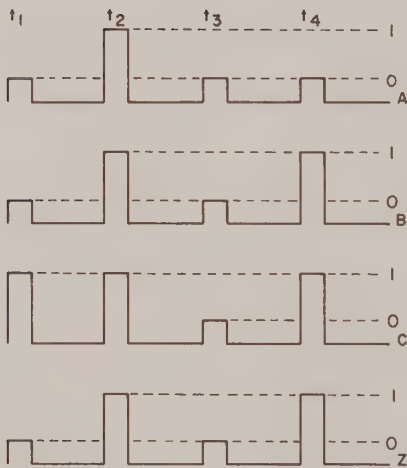
Another possible multilevel logic arrangement known as AND-OR logic is shown in Figure 14. Here, the first level consists of two AND units, each with two inputs (one input is common to both AND units). The second level consists of an OR unit. As shown, one first-level output is $A \cdot C$; the

BASIC COMBINATIONAL LOGIC

A	B	C	Z
1	1	1	1
1	1	0	0
1	0	1	1
1	0	0	0
0	1	0	0
0	0	1	0
0	1	1	1
0	0	0	0

Truth Data for AND-OR
Logic: $(A \cdot C) + (B \cdot C) = Z$

Table
10



TYPICAL TIMING DIAGRAM CONSTRUCTION
FOR $(A+B) \cdot C$ AND $(A \cdot C) + (B \cdot C)$ LOGIC
ARRANGEMENTS WHERE $(A+B) \cdot C = A \cdot C + B \cdot C$

Figure
15

other is $B \cdot C$. These outputs serve as the two inputs to the second level, resulting in a final output of $(A \cdot C) + (B \cdot C)$. This is read "A and C or B and C." The eight combinations of the three-input variables are shown in Table 10. The final output can be a logical 0 only when both inputs to the second level are logical 0's. Both of these inputs will be logical 0 if C is a logical 0 (completing the second, fourth, fifth and last rows of the table) or if A and B are both logical 0's at the same time (completing the sixth row of the table). All other combinations will produce logical 1 outputs.

A comparison of Table 10 and Table 8 will show the two to be identical. Hence, it must be true that:

$$(A + B) \cdot C = (A \cdot C) + (B \cdot C) \quad (\text{EXPANDED FORM}) \quad (9)$$

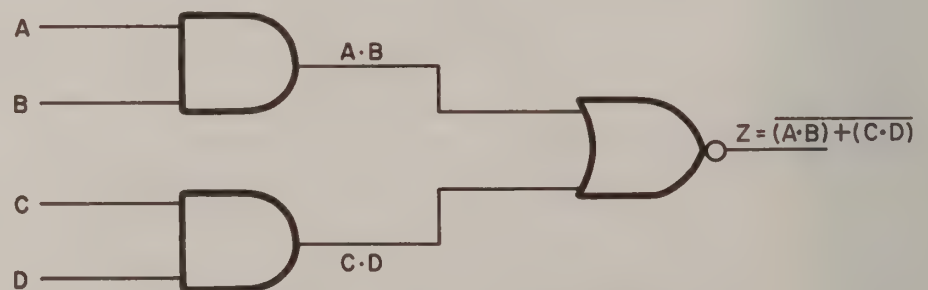
From this, it can be seen that more than one logical network (and more than one Boolean expression) can represent identical overall logical operations. In this case, the network in Figure 13 would be preferred because it uses fewer logical elements to accomplish the given operation. Generally speaking, when you encounter an existing digital system, the logical networks will have been reduced to the **MINIMUM FORM**. For the sake of economy, few manufacturers would settle for anything less than this from their design engineers.

Based on the truth data in Table 8 (or Table 10), a typical timing diagram construction for the networks in Figures 13 and 14 is shown in Figure 15. At t_1 , A and B are 0 and C is 1, leading to a 0 output. At t_2 , all three inputs are 1, leading to a 1 output. At t_3 , all three inputs are 0, leading to a 0 output. At t_4 , A is 0 and B and C are both 1's, leading to a 1 output. In terms of switching algebra, the statement over this time interval is:

$$[(0100) + (0101)] \cdot (1101) = 0101$$

or:

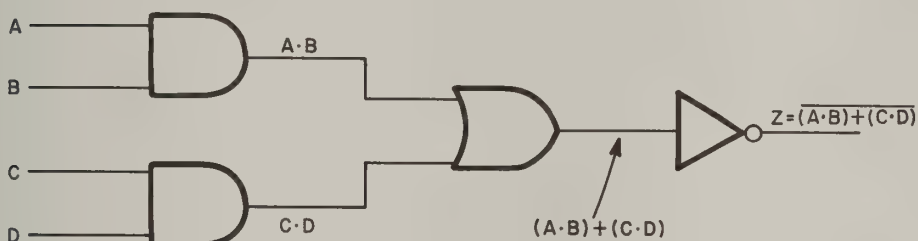
$$[(0100) \cdot (1101)] + [(0101) \cdot (1101)] = 0101.$$



THE AND-NOR LOGIC ARRANGEMENT

Figure 16

Figure 16 presents a multilevel logic AND-NOR arrangement. The first level is composed of two AND gates, each with two inputs. The two inputs are $(A \cdot B)$ and $(C \cdot D)$. These serve as inputs to the second level. Since the second level is a NOR gate, the result is $(A \cdot B) + (C \cdot D)$. The logic may also be shown as a three-level arrangement (Figure 17) and may be termed AND-OR-INVERT logic. Here, the first-level outputs are $(A \cdot B)$ and $(C \cdot D)$. When these pass through the second-level OR gate, the result is $(A \cdot B) + (C \cdot D)$. The third level performs as inversion to produce $\overline{(A \cdot B) + (C \cdot D)}$.



THE AND-OR-INVERT LOGIC INTERPRETATION OF FIGURE 16

Figure 17

Table 11 provides the truth data for these arrangements. There are $2^4 = 16$ possible input combinations. In terms of two-level logic (Figure 16), the NOR gate will produce a logical 1 output only when its two inputs are both at logical 0's at the same time. Since the first level is composed of two AND gates, the first-level outputs will be at 0 any time that either A or B are at 0 in conjunction with either C or D being at 0. This corresponds to the first three rows and to the fifth, seventh, eighth, ninth, fourteenth and fifteenth rows of Table 11, resulting in a logical 1 output, as shown. All other outputs are at 0. A typical timing diagram is shown in Figure 18. At t_1 , both B and C are 0, resulting in a logical 1 output. At t_2 , none of the inputs are 0; therefore, the output must be a logical 0. At t_3 , since B, C and D are all 0, the output is a logical 1. At t_4 , although C and D are both 0, neither A nor B is 0; this leads to a logical 0 output. Hence:

$$[(1111) \cdot (0101)] + [(0100) \cdot (1100)] = 1010$$

over the time interval shown.

The variety of basic logic combinations to produce multilevel systems is endless. However, many such combinations would perform overall operations redundant to the operations of much more simple networks. In order to deal with any combinational system which might be encountered, you must learn to develop the switching functions, truth tables and timing diagrams on a level-by-level basis so that intermediate points in the system can be tested, along with the overall system output-input relationships.

$$\overline{A \cdot B} \cdot \overline{C \cdot D} =$$

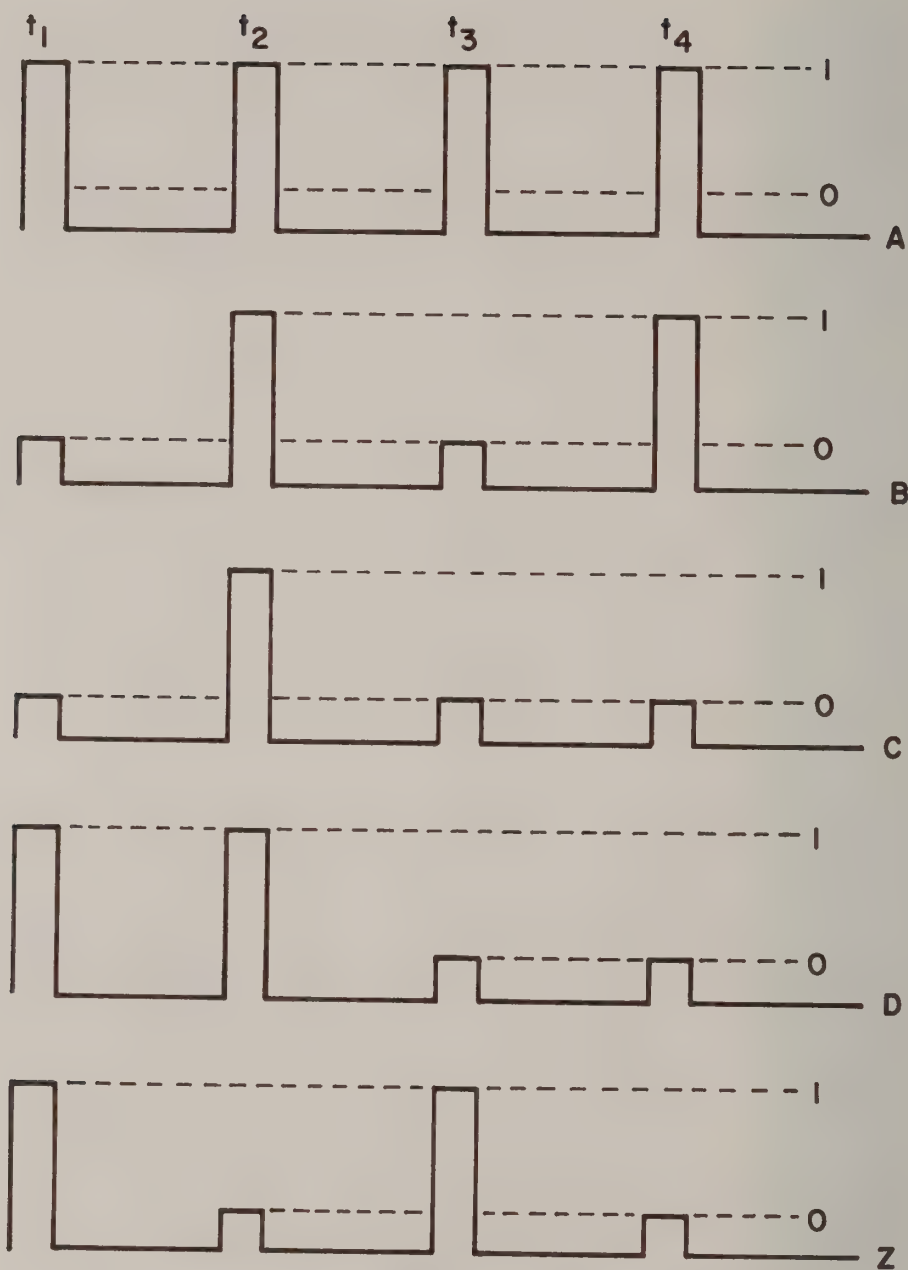
$$\overline{A+B} \cdot \overline{C+D} =$$

$$\overline{A+B} \cdot \overline{C+D} =$$

A	B	C	D	Z
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	1	0
0	1	0	1	1
1	0	0	0	1
1	0	0	1	1
1	1	0	0	0
1	0	1	1	0
1	1	1	0	0
1	1	0	1	0
1	0	1	0	1
0	1	1	0	1
1	1	1	1	0

Truth Data for:
 $\overline{(A \cdot B) + (C \cdot D)} = Z$

Table
11



TYPICAL TIMING DIAGRAM CONSTRUCTION
FOR THE $(A \cdot B) + (C \cdot D)$ LOGIC ARRANGEMENT

Figure 18

As a last example, Figure 19 presents a three-level arrangement with five input conditions. The first level contains a three-input AND gate, a two-input AND gate (with one input common) and a two-input OR gate (with one input common).

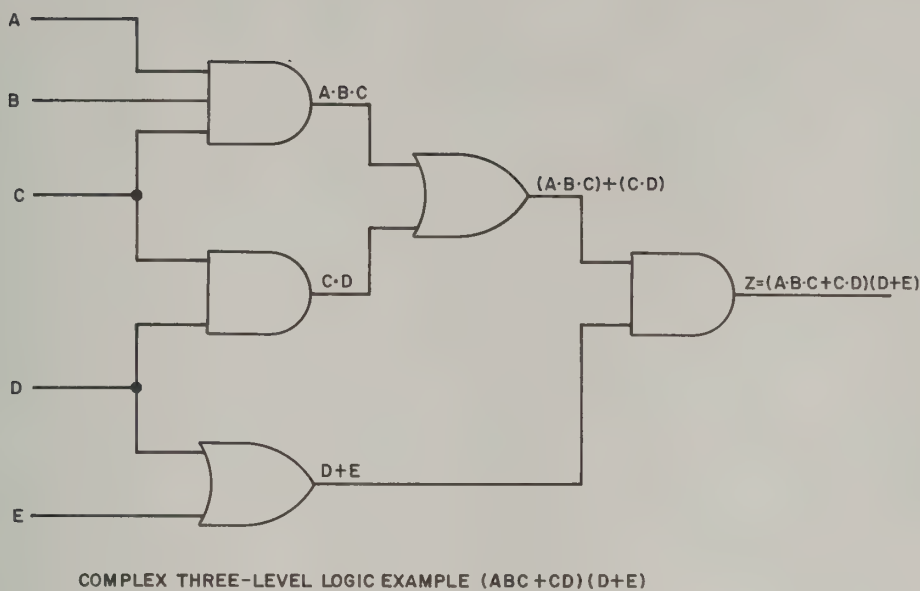


Figure 19

The three outputs from this level are $(A \cdot B \cdot C)$, $(C \cdot D)$ and $(D + E)$. The first two form the inputs for the second-level OR gate, resulting in a second-level output of $(A \cdot B \cdot C) + (C \cdot D)$. This and $(D + E)$ are applied to the inputs of the third-level function, which is an AND gate. The overall function is $(A \cdot B \cdot C + C \cdot D) \cdot (D + E)$, read as "A and B and C or C and D with D or E anded."

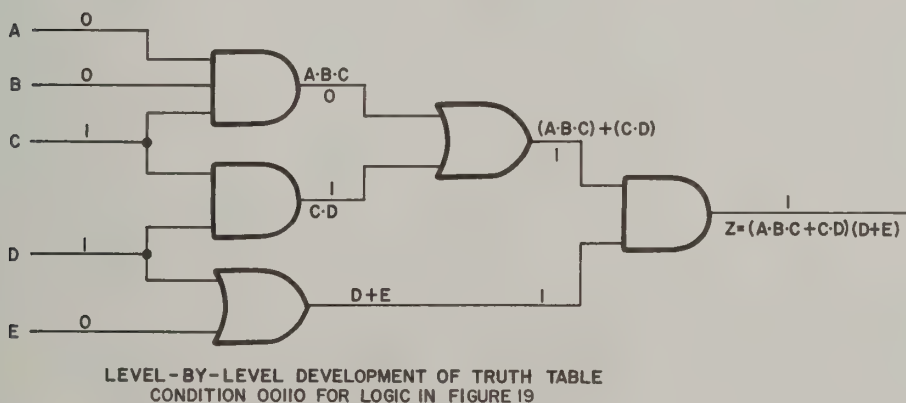


Figure 20

The truth data is presented in Table 12. There are five input variables, and thus $2^5 = 32$ possible input combinations. The output conditions can be derived through a level-by-level consideration of each possible input combination. For example, consider the instantaneous input condition 00110 (fourth row in Table 12). As shown in Figure 20, the first-level output for

A	B	C	D	E	Z
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	0	1
0	1	0	0	0	0
0	1	1	1	0	1
0	1	0	1	0	0
1	0	0	0	0	0
1	0	0	1	0	0
1	1	0	0	0	0
1	0	1	1	0	1
1	1	1	0	0	0
1	1	0	1	0	0
1	0	1	0	0	0
1	1	1	1	0	1
0	0	0	0	1	0
0	0	0	1	1	0
0	0	1	0	1	0
0	0	1	1	1	1
0	1	0	0	1	0
0	1	1	1	1	1
0	1	0	1	1	0
1	0	0	0	1	0
1	1	0	0	1	0
1	0	1	1	1	1
1	1	1	0	1	1
1	1	0	1	1	0
1	0	1	0	1	0
0	1	1	0	1	0
1	1	1	1	1	1

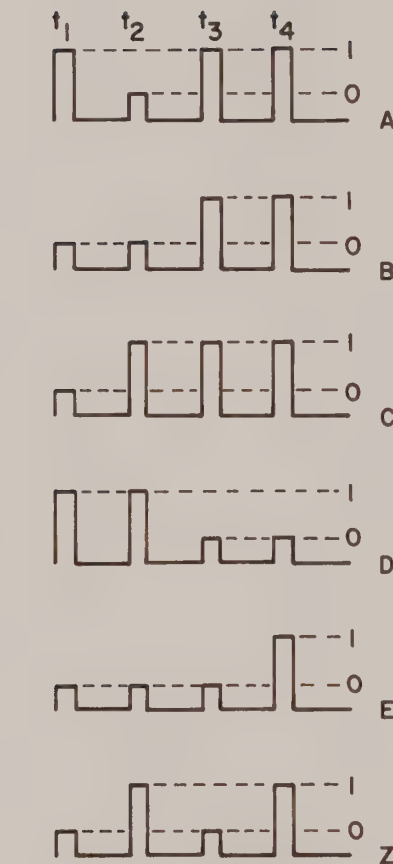
Truth Data for Multilevel Logic
in Figure 19:

$$(A \cdot B \cdot C + C \cdot D) \cdot (D + E) = Z$$

Table
12

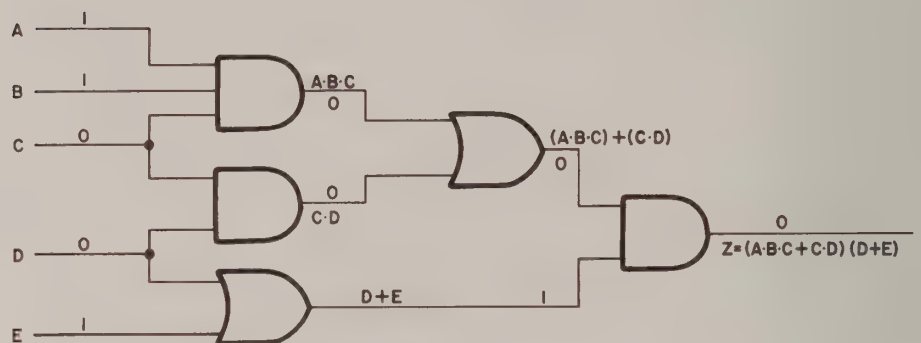
$A \cdot B \cdot C = 0 \cdot 0 \cdot 1$ is 0; for $C \cdot D = 1 \cdot 1$, the output is at 1; and for $D + E = 1 + 0$, the output is at 1. With inputs of 0 and 1, the second-level OR gate output is at 1. With inputs of 1 and 1, the third-level output is at 1. Hence, at the instant that the input condition 00110 occurs, the output will be at 1:

$$[(0 \cdot 0 \cdot 1) + (1 \cdot 1)] (1 + 0) = 1.$$



TYPICAL TIMING DIAGRAM
DEVELOPMENT FOR $(ABC + CD)$
 $(D + E)$ LOGIC ARRANGEMENT

Figure
22



LEVEL-BY-LEVEL DEVELOPMENT OF TRUTH TABLE
CONDITION 11001 FOR LOGIC IN FIGURE 19

Figure 21

A similar development of the instantaneous condition 11001 is shown in Figure 21. (This is the twenty-sixth row in Table 12.) As shown, the first-level output for $A \cdot B \cdot C = 1 \cdot 1 \cdot 0$ is 0; for $C \cdot D = 0 \cdot 0$, the output is 0; and for $D + E = 0 + 1$, the output is 1. With inputs of 0 and 0, the second-level output is 0. With inputs of 0 and 1, the third-level output is 0. Hence, at the instant that the input condition 11001 occurs, the output will be 0:

$$[(1 \cdot 1 \cdot 0) + (0 \cdot 0)] (0 + 1) = 0.$$

This procedure can be applied to every input combination in Table 12 (or to every input combination for any other multileveled logic network).

Based on the truth data in Table 12, the timing diagram of Figure 22 can be developed for the indicated sequences of input digits. You should carefully examine each time interval, justifying the indicated output condition of each.

THE EXCLUSIVE-OR FUNCTION

The OR function:

$$A + B = Z$$

The following Practice Exercise questions cover the subjects which you have just studied. They are:

TIMING DIAGRAMS

MULTILEVEL LOGIC ARRANGEMENTS

MULTIPLE-INPUT MULTILEVEL LOGIC

1. Boolean expressions (or switching functions), truth tables, and timing diagrams are all different forms of expressing the output-input relationships for digital systems. True or False?
2. In Figure 3C, why is the output condition at logical 0 during the interval prior to the instant that A and B both go to logical 1?
3. How can the truth table for a logical unit be used as an aid to constructing the timing diagram for this unit?
4. How does the logical digit representation of the signals for a short-duration pulse system differ from that of a more "static" system?
5. Using a convenient time interval scale, draw the digital signal associated with the sequence: 100110.
6. Using a convenient time interval scale, draw the complete timing diagram for the operation shown below.



7. A two-level arrangement consisting of AND-NOT will produce the NAND function. True or False?
8. Draw the two-level logic diagram for NOR-NOT and indicate the overall function for this arrangement, derived on a level-by-level basis.
9. Prove that $\overline{\overline{A + B}} = \overline{A \cdot B} = Z$ by constructing the level-by-level truth table (similar to Table 5) and selecting those parts that represent the overall circuit operation.
10. Describe how the logic network in Figure 13 produces the output $(A + B) \cdot C$.
11. Describe how the third row of Table 8 is formed.

12. Why do the last four rows in Table 9 produce a logical 0 output?
13. Different logic expressions result from different network arrangements and always provide different overall functions. True or False?
14. Logic networks have been reduced to their minimum form when the desired function is accomplished with the least possible number of basic logic units. True or False?
15. Using either a timing diagram or the appropriate truth table, determine the digital output sequence for the networks in Figures 13 and 14 when $A = 1011$, $B = 1010$, and $C = 0010$.
16. Describe the development of the level-by-level logic in Figure 17.
17. Justify the development of the logical 1 output in the fifth row of Table 11 on a level-by-level basis using the logic in Figure 16.
18. If the second-interval digit for input A in Figure 18 were a logical 0, what would the output digit sequence be?
19. Describe the development of the input conditions for the third level of the logic network in Figure 19.

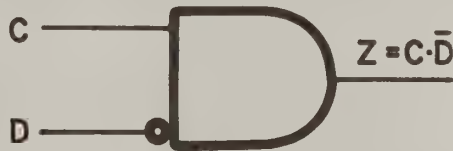
is treated as an INCLUSIVE-OR operation because either A or B or both A and B (simultaneously) will produce a logical 1 output. In a more complete form, the INCLUSIVE-OR function would be written as:

$$A + B + (A \cdot B) = Z.$$

However, this is usually not necessary. A second form of the OR function is known as **EXCLUSIVE-OR**. This form does not produce an output at logical 1 when A and B are both at 1. It produces an output only when either A or B is at 1. This is written as:

$$(A + B) \overline{(A \cdot B)} = Z \quad (\text{EXCLUSIVE-OR}) \quad (10)$$

which is read "A or B and A and B notted" or "A or B, but not both." This is the OR-AND form of EXCLUSIVE-OR.



THE INHIBIT FUNCTION
SYMBOL

Figure 23

The "but not" nature of the EXCLUSIVE-OR operation implies the presence of an INHIBITING function. That is, a logical 1 at A or at B is allowed to produce a logical 1 output, but if logical 1's at A and at B occur together, the output must be inhibited (a logical 0). The basic INHIBIT function symbol is shown in Figure 23. This indicates that it is similar to an AND gate, where one terminal C behaves as a positive logic arrangement (called the signal input), while the other D (which is called the inhibit input) behaves as a negative logic arrangement. The inhibit input causes an inversion of the condition at terminal D. Hence, the logic is:

$$C \cdot \overline{D} = Z \quad (\text{INHIBIT}) \quad (11)$$

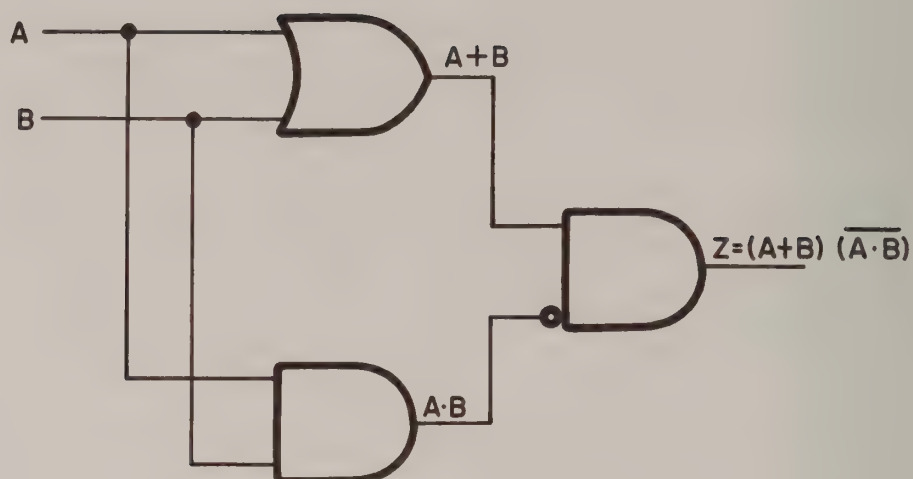
The truth table is presented as Table 13. Since there are two input terminals, there are four possible conditions. Notice that any time a 1 occurs at the INHIBIT terminal (D), the output must be at logic 0.

The nature of the EXCLUSIVE-OR expression suggests that this operation must be accomplished with a multilevel logic arrangement. Such an arrangement is shown in Figure 24. The first level consists of an OR gate and an AND gate with common inputs. The second level is an INHIBIT gate. The

Signal Input Point	Inhibit Input Point	Output Point
C	D	Z
1	1	0
1	0	1
0	1	0
0	0	0

Truth Data for INHIBITING
Function $C \cdot \overline{D} = Z$

Table
13



AN EXCLUSIVE-OR LOGIC ARRANGEMENT (OR-AND FORM)

Figure 24

first-level OR gate output is $A + B$; the other AND gate output is $A \cdot B$. The first output is fed to the positive logic terminal of the INHIBIT gate and the second output is fed to the negative logic terminal. Hence, the second signal is inverted while the first is not, to produce $(A + B) \cdot (\overline{A \cdot B})$.

A	B	Z
1	1	0
1	0	1
0	1	1
0	0	0

Truth Data for
EXCLUSIVE-OR:
 $(A + B)(\overline{A \cdot B}) = Z$

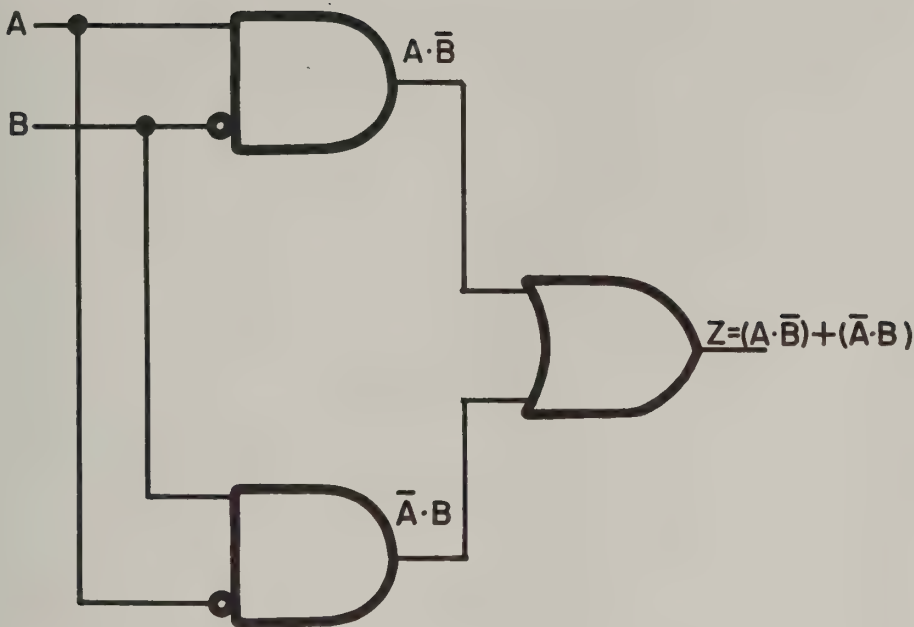
Table
15

First-Level Inputs		First-Level Outputs (Second-Level Inputs)		Final Output
A	B	$(A + B)$	$(A \cdot B)$	Z
1	1	1	1	0
1	0	1	0	1
0	1	1	0	1
0	0	0	0	0

Level-by-Level Construction of Truth Data for Logic
in Figure 24

Table 14

The truth table construction is shown, level by level, in Table 14. The first-level output $(A + B)$ is at 0 only when both A and B are at 0. The first-level output $(A \cdot B)$ is at 1 only when both A and B are at 1. These two sets of conditions are inputs to the INHIBIT gate. Using the conditions shown in Table 13, and thinking of $(A + B)$ as C and $(A \cdot B)$ as D, the final output can be constructed. Selecting the two original input columns and the final output column from Table 14, the EXCLUSIVE-OR truth table appears as in Table 15.



AN ALTERNATE EXCLUSIVE-OR LOGIC ARRANGEMENT (AND-OR FORM)

Figure 25

A second form of the EXCLUSIVE-OR logic is shown in Figure 25. Here, the first level consists of two INHIBIT gates with common, “crossed” inputs. The two first-level outputs are $A \cdot \bar{B}$ and $\bar{A} \cdot B$, from the upper and lower INHIBIT gates, respectively. With these as inputs to an OR gate, the final output is:

$$(A \cdot \bar{B}) + (\bar{A} \cdot B) = Z \quad (\text{EXCLUSIVE-OR}) \quad (12)$$

This is an alternate for the EXCLUSIVE-OR expression known as the AND-OR form. If A and B are both logical 1, the two first-level outputs are both 0 (A inhibits the bottom unit; B inhibits the top unit). This produces a 0 from the second-level OR gate. If A and B are both 0, of course, the final output is still 0. However, if either A or B is 1 while the other is 0, one unit in the first level is inhibited; the other is not. Hence, the second-level OR gate will “see” at least one logical 1 at its input, resulting in a 1 output, and Table 15 still applies.

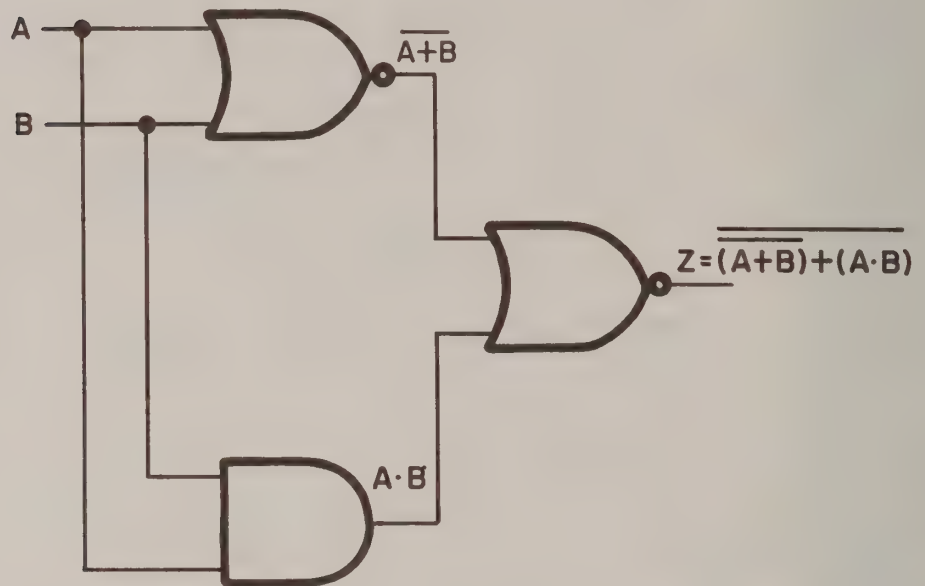
The EXCLUSIVE-OR function is used so frequently in digital systems that the multilevel logic of either Figure 24 or Figure 25 (or any other logic providing the equivalent operations) is generally replaced with the composite symbol shown in Figure 26. From a system viewpoint, it is of little consequence whether the function is accomplished with the OR-AND network or



THE COMPOSITE EXCLUSIVE-
OR SYMBOL

Figure
26

the AND-OR network or any other network. The important point is that the EXCLUSIVE-OR output-input relationships be maintained, as expressed by Table 15. The same is true of any other complex logic function.



AN ALTERNATE EXCLUSIVE-OR ARRANGEMENT NOT USING THE INHIBIT FUNCTION

Figure 27

Figure 27 illustrates another arrangement for performing the EXCLUSIVE-OR function. This circuit does not use the INHIBIT function described previously. Rather, it uses the two-level logic, with the first level consisting of a NOR gate and an AND gate with common inputs. The outputs from this level are $\overline{A+B}$ and $A \cdot B$. With these as inputs, the second-level NOR gate provides the output $\overline{(\overline{A+B}) + (A \cdot B)}$.

The truth table development is shown, level by level, in Table 16. When A and B are both logical 1, the first-level outputs are 0 and 1. With these as inputs, the second level provides a 0 output. When either A or B are logical 1, the first-level outputs are 0 and 0. With these as inputs, the second level provides a 1 output. When A and B are both 0, the first-level outputs are 1 and 0 and the second level provides a 0 output. The input and output columns of Table 16 can be seen to be identical to Table 15, justifying the fact that:

$$\overline{(\overline{A+B}) + (A \cdot B)} = (A+B) \cdot \overline{(A \cdot B)} = Z \quad (\text{EXCLUSIVE-OR}) \quad (13)$$

First-Level Inputs		First-Level Outputs (Second-Level Inputs)		Final Output
A	B	$(\overline{A + B})$	$(A \cdot B)$	Z
1	1	0	1	0
1	0	0	0	1
0	1	0	0	1
0	0	1	0	0

Level-by-Level Construction of Truth Data for Logic in Figure 27

Table 16

and that the logic in Figure 27 is EXCLUSIVE-OR. [This equivalence (and all other equivalences in this lesson) can also be derived mathematically. We will describe these techniques in the next section.]

It is also possible to perform an inversion of the EXCLUSIVE-OR function, known as EXCLUSIVE-NOR. In Figure 27, the removal of the NOT part of the output NOR function (changing the second-level circuit to an OR gate) changes the entire circuit from an EXCLUSIVE-OR to an EXCLUSIVE-NOR gate, where the output is $Z = (\overline{A + B}) + A \cdot B = \overline{A} \cdot \overline{B} + A \cdot B$. The inversion or complementation of $(A \cdot B) + (\overline{A} \cdot \overline{B})$ produces the function $(A \cdot B) + (\overline{A} \cdot \overline{B})$, which is one possible form of EXCLUSIVE-NOR. (Any other form of the EXCLUSIVE-OR function may also be complemented to obtain alternate expressions; they are all equivalent in terms of the overall operation performed.)

The EXCLUSIVE-NOR truth table may be obtained by complementing the outputs in Table 15. The result is shown in Table 17. An examination of the output-input conditions shown in this table will indicate that the output is 1 when A and B are both 0, or when A and B are both 1. Hence, a simple Boolean expression for EXCLUSIVE-NOR is $\overline{A} \cdot \overline{B} + A \cdot B$. This form is usually used to identify the EXCLUSIVE-NOR operation since:

$$(\overline{A \cdot B}) + (A \cdot B) = \overline{A} \cdot \overline{B} + A \cdot B = Z \quad (\text{EXCLUSIVE-NOR}) \quad (14)$$

A	B	Z
1	1	1
1	0	0
0	1	0
0	0	1

Truth Data for EXCLUSIVE-NOR

Table 17



THE COMPOSITE EXCLUSIVE-NOR SYMBOL

Figure
28

A	B	Z
0	0	0
0	1	1
1	0	1
1	1	0

Truth Data for a
Two-variable ProcessTable
18

A	B	C	Z
1	1	1	1
1	0	1	1
0	1	1	1
0	0	1	0
1	1	0	0
1	0	0	0
0	1	0	0
0	0	0	0

Truth Data for a
Three-variable ProcessTable
19

In fact, any other expression which may be derived for other EXCLUSIVE-NOR networks can be reduced to the form $\bar{A} \cdot \bar{B} + A \cdot B$. The composite symbol for EXCLUSIVE-NOR is shown in Figure 28.

DESIGNING COMBINATIONAL LOGIC

When a specific set of complex logical operations is to be performed, the necessary logic arrangement must be “created.” That is, the desired operations—expressed in plain English, a truth table, or a Boolean equation—must be developed into a functional arrangement consisting of the basic “logic blocks.”

The first step in designing a combinational arrangement is the conversion of a set of technical statements which relate the input conditions to the desired output actions. For example, a certain industrial process may have two controlling conditions, identified by A and B. The output (Z) must satisfy the following relationships:

- (1) When A and B are both 0, Z must be 0;
- (2) When A is 0 and B is 1, Z must be 1;
- (3) When A is 1 and B is 0, Z must be 1; and
- (4) When A and B are both 1, Z must be 0.

These “verbal” statements are shown in truth table form by Table 18. Each row in the table corresponds to one of the statements, as shown by the column at the left. This truth table provides a complete statement of all logical relationships which must be satisfied by the logic arrangement. (You may recognize this as the data for the EXCLUSIVE-OR function. However, this recognition is not essential at the moment.)

As a second example, consider a process where a single output (Z) will be controlled by three input conditions, identified by A, B and C. The following relationships have been specified:

- (1) Z can be 1 only when C is 1; and
- (2) When C is 1, if A and B are both 0, Z will be 0, otherwise Z will be 1.

This set of relationships is not as obvious in truth table form as the previous example. However, since there are three inputs, there must be $2^3 = 8$ rows in the truth table. Furthermore, half of these will have C equal to 1; the other

half, C equal to 0. (And each half will have A and B in every possible combination that they can achieve.) This is shown in Table 19. From the first statement (by inference), the half of the table with C equal to 0 can be completed with Z equal to 0. In the half of the table with C equal to 1, the row with both A and B equal to 0 can be completed—Z must be 0 (fourth row) according to the second statement. From this same statement, all remaining values of Z must be 1.

As a last example, consider a process controlled by four input conditions, identified as A, B, C and D. There are two outputs, identified as Z_1 and Z_2 . The output Z_1 must satisfy the following requirements:

- (1) Regardless of the value of C and D, Z_1 must be 1 when A and B are both 0; and
- (2) Regardless of the value of A and B, Z_1 must be 1 when C and D are both 0.

The output Z_2 must satisfy the following statements:

- (1) Z_2 can be 1 only when C and D are both 0; and
- (2) When C and D are both 0, Z_2 will be 0 if A and B are both 1 —all other values of A and B will cause Z_2 to be 1.

In order to develop the truth data for these conditions, the two outputs should be considered independently. Since there are four inputs, both tables will contain $2^4 = 16$ rows.

The data for Z_1 are shown in Table 20. Half of the rows in this table must have D equal to 1; in the other half, D must equal 0. Of each of these halves, one-half of the rows must have C equal to 1; for the other half, C must equal 0. This structure is shown in Table 20. The remaining entries are made up of the various combinations of A and B. From the first statement for Z_1 , all rows with A and B equal to 0 (regardless of the value of C and D) can be completed by setting Z_1 equal to 1 (the fourth row and every fourth row). From the second statement for Z_1 , all rows with C and D equal to 0 (regardless of the value of A and B) can be completed by setting Z_1 equal to 1 (in the last four rows). All of the other entries can be set to $Z_1 = 0$.

The data for Z_2 are shown in Table 21. The input combinations are identical to those of Table 20. From the first statement for Z_2 , all rows with both C and D not 0 can be completed by setting Z_2 equal to 0. From the second statement for Z_2 , the remaining rows, which have A and B both equal to 1, can be completed by setting Z_2 equal to 0. The rest of the entries have Z_2 equal to 1.

A	B	C	D	Z_1
1	1	1	1	0
1	0	1	1	0
0	1	1	1	0
0	0	1	1	1
1	1	0	1	0
1	0	0	1	0
0	1	0	1	0
0	0	0	1	1
1	1	1	0	0
1	0	1	0	0
0	1	1	0	0
0	0	1	0	1
1	1	0	0	1
1	0	0	0	1
0	1	0	0	1
0	0	0	0	1

Truth Data for Z_1 with Given Conditions

Table
20

A	B	C	D	Z_2
1	1	1	1	0
1	0	1	1	0
0	1	1	1	0
0	0	1	1	0
1	1	0	1	0
1	0	0	1	0
0	1	0	1	0
0	0	0	1	0
1	1	1	0	0
1	0	1	0	0
0	1	1	0	0
0	0	1	0	0
1	1	0	0	0
1	0	0	0	1
0	1	0	0	1
0	0	0	0	1

Truth Data for Z_2 with Given Conditions

Table
21

Switching Functions from Truth Tables

Once a truth table has been developed for a set of complex logical operations, it is possible to derive Boolean expressions or switching functions which also represent these operations. Notice that more than one expression is implied. This is deliberate. There is an almost endless variety of Boolean expressions that can be derived for any given set of logical operations. They are all equivalent, in that they have the same truth data and serve to accomplish the same function. However, each expression represents a different logic network for accomplishing the function. The importance of this aspect will be described later. The EXCLUSIVE-OR function provides an excellent example of this "logical ambiguity."

Although the EXCLUSIVE-OR is a common function, it is not a simple basic function in the sense of AND, OR, NOT, NAND and NOR. Rather, it is a combination of several of these basic functions. In fact, it may be accomplished by any one of several combinations of these functions. The fundamental expression for EXCLUSIVE-OR is given by Equation (10) in the preceding topic. The logic arrangement for this expression is provided in Figure 24. However, this same topic shows two additional switching functions—Equations 12 and 13—which are also the EXCLUSIVE-OR operation, but with different logic arrangements (Figures 25 and 27). Even these three alternatives do not exhaust the various EXCLUSIVE-OR functions and arrangements; many more are possible. This is generally true of all combinational logic.

A	B	Z	Complementary OR- Interpretation of Output 0's	AND-Interpretation of Output 1's
0	0	0	$A + B$	Not Used
0	1	1	Not Used	$\bar{A} \cdot B$
1	0	1	Not Used	$A \cdot \bar{B}$
1	1	0	$\bar{A} + \bar{B}$	Not Used

Initial Step for Converting Entries in EXCLUSIVE-OR
Truth Table to Boolean Expressions

Table 22

There are two alternate approaches for interpreting the entries in a truth table as switching functions. Both of these are shown in Table 22 (using the EXCLUSIVE-OR data from Table 15 as a "starting point"). One approach assumes that the input conditions resulting in output zeros represent OR functions and replaces these input digits with complementary Boolean symbols, such as A and B for input 0's; $\bar{A}$ and $\bar{B}$ for input 1's. The second approach assumes that the input conditions resulting in output 1's represent AND functions and replaces these input digits directly with Boolean symbols (A and B for input 1's; and $\bar{A}$ and $\bar{B}$ for input 0's).

Using the complementary OR-interpretation of output 0's in Table 22, we can combine the two terms to form a **LOGICAL PRODUCT**. Hence:

$$(A + B) \cdot (\bar{A} + \bar{B}) = Z$$

results directly from the data in Table 22. This interpretation is known as a logical product, OR-AND expression, or as the **PRODUCT OF SUMS** form of the switching function. Using the AND-interpretation of output 1's in Table 22, a set of expressions known as the **SUM OF PRODUCTS** form, or an AND-OR expression, is obtained. We combine these two terms to form a **LOGICAL SUM**. Hence:

$$(A \cdot \bar{B}) + (\bar{A} \cdot B) = Z$$

can be obtained directly from Table 22.

Only one of these expressions:

$$(A \cdot \bar{B}) + (\bar{A} \cdot B) = Z$$

matches an expression for an EXCLUSIVE-OR circuit described previously (Figure 25). The other expression represents an entirely different logic arrangement, but still represents the EXCLUSIVE-OR function.

Table 23 presents the truth data of Table 19 in terms of the complementary OR-interpretation of output 0's, and interpretation of output 1's for the individual rows. The logical product (or product of sums form) for Z is given by:

$$(A + B + \bar{C}) \cdot (\bar{A} + \bar{B} + C) \cdot (A + B + C) \cdot (A + \bar{B} + C) \cdot (\bar{A} + B + C) = Z$$

Complementary OR- Interpretation of Output 0's	AND-Interpretation of Output 1's
Not Used	$A \cdot B \cdot C$
Not Used	$A \cdot \bar{B} \cdot C$
Not Used	$\bar{A} \cdot B \cdot C$
$A + B + \bar{C}$	Not Used
$\bar{A} + \bar{B} + C$	Not Used
$\bar{A} + B + C$	Not Used
$A + \bar{B} + C$	Not Used
$A + B + C$	Not Used

Initial Step for Obtaining Boolean Interpretation
of Entries in Table 19

Table 23

Complementary OR- Interpretation of Output 0's	AND-Interpretation of Output 1's
$\bar{A} + \bar{B} + \bar{C} + \bar{D}$	Not Used
$\bar{A} + B + \bar{C} + \bar{D}$	Not Used
$A + \bar{B} + \bar{C} + \bar{D}$	Not Used
Not Used	$\bar{A} \cdot \bar{B} \cdot C \cdot D$
$\bar{A} + \bar{B} + C + \bar{D}$	Not Used
$\bar{A} + B + C + \bar{D}$	Not Used
$A + \bar{B} + C + \bar{D}$	Not Used
Not Used	$\bar{A} \cdot \bar{B} \cdot \bar{C} \cdot D$
$\bar{A} + \bar{B} + \bar{C} + D$	Not Used
$\bar{A} + B + \bar{C} + D$	Not Used
$A + \bar{B} + \bar{C} + D$	Not Used
Not Used	$\bar{A} \cdot \bar{B} \cdot C \cdot \bar{D}$
Not Used	$A \cdot B \cdot \bar{C} \cdot \bar{D}$
Not Used	$A \cdot \bar{B} \cdot \bar{C} \cdot \bar{D}$
Not Used	$\bar{A} \cdot B \cdot \bar{C} \cdot \bar{D}$
Not Used	$\bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \bar{D}$

Initial Step in Obtaining Boolean Interpretation
of Entries in Table 20

Table
24

using the rows from the complementary-OR interpretation of output 0's. The sum of products form is obtained by using the rows from the AND-interpretation of output 1's, resulting in the logical sum:

$$(A \cdot B \cdot C) + (A \cdot \bar{B} \cdot C) + (\bar{A} \cdot B \cdot C) = Z.$$

Table 24 presents the truth data for Z_1 from Table 20 in terms of the two interpretations that have been described. Using the complementary OR-interpretations of the output 0's, the product of sums form for Z_1 is:

$$\begin{aligned} & (\bar{A} + \bar{B} + \bar{C} + \bar{D}) \cdot (\bar{A} + B + \bar{C} + \bar{D}) \cdot (A + \bar{B} + \bar{C} + \bar{D}) \\ & \cdot (\bar{A} + \bar{B} + C + \bar{D}) \cdot (\bar{A} + B + C + \bar{D}) \cdot (A + \bar{B} + C + \bar{D}) \\ & \cdot (\bar{A} + \bar{B} + \bar{C} + D) \cdot (\bar{A} + B + \bar{C} + D) \cdot (A + \bar{B} + \bar{C} + D) = Z_1. \end{aligned}$$

The sum of products form is written from the rows, using the AND-interpretation of output 1's:

$$\begin{aligned} & (\bar{A} \cdot \bar{B} \cdot C \cdot D) + (\bar{A} \cdot \bar{B} \cdot \bar{C} \cdot D) + (\bar{A} \cdot \bar{B} \cdot C \cdot \bar{D}) \\ & + (A \cdot B \cdot \bar{C} \cdot \bar{D}) + (A \cdot \bar{B} \cdot \bar{C} \cdot \bar{D}) \\ & + (\bar{A} \cdot B \cdot \bar{C} \cdot \bar{D}) + (\bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \bar{D}) = Z_1. \end{aligned}$$

In the previous examples the number of components in the product of sums expression plus the number of components in the sum of products expression is equal to the number of entries in the originating truth table (2^n , where n is the number of inputs). In the first example, each expression has two components. In the second example, the product of sums form has five components and the sum of products form has only three components. Here, the sum of products expression is the simplest. In the third example, the product of sums form has nine components and the sum of products form has seven components. Again, the sum of products expression is the simplest. Usually, when we derive a Boolean expression from a truth table, we use only the simplest form. We have shown both forms here so you can become familiar with the alternate approaches.

As a last example, we will derive the simplest Boolean expression for Z_2 from the truth data in Table 21. Since there are 13 zero outputs and only 3 one outputs in this table, we will obtain the simplest expression by using the AND-interpretation of the output 1's. The result is shown in Table 25. Using these components, we obtain the sum of products form:

$$(A \cdot \bar{B} \cdot \bar{C} \cdot \bar{D}) + (\bar{A} \cdot B \cdot \bar{C} \cdot \bar{D}) + (\bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \bar{D}) = Z_2.$$

AND-Interpretation of Output 1's
$A \cdot \bar{B} \cdot \bar{C} \cdot \bar{D}$
$\bar{A} \cdot B \cdot \bar{C} \cdot \bar{D}$
$\bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \bar{D}$

Initial Step in Obtaining
Sum of Products
Boolean Interpretation
of Entries in Table 21

Table
25

Simplification of Switching Functions

Multiple expressions have been described for the same logical operation. Expressions can always be shown to be equivalent or not equivalent by constructing a set of truth tables, one for each expression. Expressions with the same truth table are logically equivalent. With a background in switching algebra, you can rearrange switching functions in an algebraic manner. This allows multiple expressions to be rewritten to check their equivalency. With a basic knowledge of switching algebra, it is also possible to greatly simplify the **EXPANDED FORM** of switching function. This is very important if you are to minimize the number of logic components needed to implement a given operation.

Table 26 lists some of the basic switching algebraic relationships used to simplify expressions. Relationships 1 and 2 show that $A + A$ and $A \cdot A$ are redundant forms; they are simply equivalent to A . Relationships 3 and 4 show that $A + \bar{A}$ and $A \cdot \bar{A}$ are independent forms. That is, $(A + \bar{A})$ is always equal to 1; because, if $A = 1$, then $\bar{A} = 0$ and we have $(1 + 0) = 1$. However, when $A = 0$, then $\bar{A} = 1$ and we have $(0 + 1) = 1$. Similarly, $(A \cdot \bar{A})$ is always equal to 0, since A and $\bar{A}$ cannot both be 1. Relationships 5 and 6 show that the order of a logical sum or logical product is of no consequence. Relationship 7 shows that a logical product with one OR-term and a single "multiplier" can be treated like a normal algebraic expansion. However, relationship 8 shows that a logical product of two OR-terms does not behave in a normal algebraic fashion. Relationships

1. $A + A = A$	2. $A \cdot A = A$
3. $A + \bar{A} = 1$	4. $A \cdot \bar{A} = 0$
5. $A + B = B + A$	6. $A \cdot B = B \cdot A$
7. $A \cdot (B + C) = (A \cdot B) + (A \cdot C)$	8. $A + (B \cdot C) = (A + B) \cdot (A + C)$
9. $\bar{\bar{A}} = A$	10. $A \cdot (A + B) = A$
11. $A + (A \cdot B) = A$	12. $A \cdot (\bar{A} + B) = A \cdot B$
13. $A + (\bar{A} \cdot B) = A + B$	14. $\bar{A} \cdot (A + B) = \bar{A} \cdot B$
15. $\bar{A} + (A \cdot B) = \bar{A} + B$	16. $\bar{A} \cdot (A + \bar{B}) = \bar{A} \cdot \bar{B}$
17. $\bar{A} + (A \cdot \bar{B}) = \bar{A} + \bar{B}$	18. $A + B + \bar{A} \cdot \bar{B} = 1$
19. $A \cdot B + A \cdot \bar{B} + \bar{A} \cdot B = A + B$	20. $A \cdot B + A \cdot \bar{B} + \bar{A} \cdot B + \bar{A} \cdot \bar{B} = 1$

Simplification Relationships for Boolean Expressions

Table 26

10 through 20 are unique to switching algebra. Relationship 9 simply states the double negation process in algebraic form.

The function:

$$(A \cdot B) + (A \cdot \bar{B}) = Z$$

has an **A** common to both **TERMS**. Hence, it can be rewritten with the **A** "factored out":

$$A (B + \bar{B}) = Z.$$

However, $(B + \bar{B})$ is like relationship 3 in Table 26 and the output is independent of this **FACTOR**. Hence:

$$(A \cdot B) + (A \cdot \bar{B}) = A = Z.$$

The function:

$$A = Z$$

is certainly easier to implement than the function:

$$(A \cdot B) + (A \cdot \bar{B}) = Z.$$

The following Practice Exercise questions cover the subjects which you have just studied. They are:

THE EXCLUSIVE-OR FUNCTION

DESIGNING COMBINATIONAL LOGIC

SWITCHING FUNCTIONS FROM TRUTH TABLES

20. Write the OR-AND form of the EXCLUSIVE-OR function with E and F as the input conditions.
21. The INHIBIT function is 0 only when there is a 1 at the inhibiting terminal. True or False?
22. Justify the third row output in Table 15 for the logic network in Figure 24.
23. $(A + B)(\overline{A} \cdot \overline{B}) = (A \cdot \overline{B}) + (\overline{A} \cdot B)$. True or False?
24. Justify the first row output in Table 15 for the logic network in Figure 25.
25. From a system viewpoint, what is the most important aspect of any complex logic function?
26. Describe the level-by-level development of the output for the logic network in Figure 27 when $A = 1$ and $B = 0$.
27. Using any convenient time interval scale and the truth data in Table 17, develop the timing diagram for the EXCLUSIVE-NOR function when $A = 0110$ and $B = 0111$.
28. What are the three possible "starting points" for the synthesis of a functional system?
29. Describe why the half of Table 19 with C equal to 0 must also have Z equal to 0.
30. Describe why the row in Table 21 with both A and B equal to 1 and both C and D equal to 0 must have Z_2 equal to 0.
31. Construct the truth table for a process having three input conditions (A, B and C) and one output condition (Z), where the following statements apply:
 - (1) Regardless of the value of A and B, Z will be 1 if C is 1; and
 - (2) If C is 0, Z can only be 1 if A and B are both 1.
32. Different switching functions may represent the same complex logical function. True or False?

33. What are the two different ways in which the input conditions of a truth table may be interpreted as switching functions?
34. What is a product of sums expression?
35. What is a sum of products expression?
36. The total number of components in both the AND-OR and OR-AND expressions derived from a given truth table must equal the number of input combinations in the table. True or False?
37. Write the sum of products expression for Z in Table 23.
38. Write the product of sums expression for Z_1 in Table 24.
39. Write both the product of sums and sum of products expressions for Z from the truth data in the following table.

A	B	C	Z
0	0	0	0
1	0	0	1
0	0	1	1
1	1	0	0
0	1	1	0
1	0	1	1
0	1	0	0
1	1	1	0

Yet, the two are logically equivalent.

The function:

$$(A \cdot B) + (A \cdot B \cdot C) = Z$$

has $A \cdot B$ common to both terms. Hence:

$$(A \cdot B) [1 + C] = Z$$

results when “factoring” is performed. The form $1 + C$ may appear to be an “odd” arrangement. However, since C can only be a logical 1 or a logical 0, the form must have one of two values:

$$1 + C = 1 + 1 = 1$$

$$1 + C = 1 + 0 = 1$$

In either case, the output is independent of the value of C and:

$$(A \cdot B) [1 + C] = A \cdot B = Z.$$

Again, the expression:

$$A \cdot B = Z$$

is much easier to implement than the expression:

$$(A \cdot B) + (A \cdot B \cdot C) = Z.$$

Yet, the two are logically equivalent. Other forms similar to the $1 + C$ occasionally result during the process of logical manipulations. The common ones are summarized in Table 27.

1. $A + 0 = A$	2. $A \cdot 1 = A$
3. $A + 1 = 1$	4. $A \cdot 0 = 0$

Special Equivalents Occurring in
Boolean Manipulations

Table
27

The function:

$$(A \cdot B) \cdot [A + B + (\bar{A} \cdot C)] = Z$$

is best simplified by “expanding” the product:

$$(A \cdot B \cdot A) + (A \cdot B \cdot B) + (A \cdot B \cdot \bar{A} \cdot C) = Z.$$

From relationship 2 in Table 26, $A \cdot A$ and $B \cdot B$ reduce to A and B , respectively. From relationship 4 in Table 26, $A \cdot \bar{A}$ is always equal to 0. Hence, the expression reduces to:

$$(A \cdot B) + (A \cdot B) + (0 \cdot B \cdot C) = Z.$$

The first two terms are redundant (like relationship 1 of Table 26), and the third term is like relationship 4 of Table 27. Hence:

$$A \cdot B = Z.$$

There are other simplification relationships which can be developed in addition to those provided in Tables 26 and 27. However, these tables only contain the “fundamental” simplification relationships. A principle known as **DE MORGAN’S THEOREM** can also be very useful in simplifying switching functions, as well as in proving the equivalence of two such functions. This is especially true when the functions contain several negation or complementation levels. The principle is discussed in an Appendix to this lesson.

To prove relationship 13 you can first expand it:

$$A [1] + (\bar{A} \cdot B) = A \cdot (\bar{B} + B) + \bar{A} \cdot B = A \cdot \bar{B} + A \cdot B + \bar{A} \cdot B$$

(sum of products).

Now each term has an A or an $\bar{A}$ and a B or a $\bar{B}$. Note that $A \cdot B$ has an A in common with $A \cdot \bar{B}$, while $A \cdot B$ also has a B in common with $\bar{A} \cdot B$. Since $A + A = A$ (relationship 1), we can use $A \cdot B$ twice:

$$A \cdot B + A \cdot B = A \cdot B.$$

$A \cdot B + A \cdot B$ is substituted into the sum-of-products giving

$$\begin{aligned} A \cdot \bar{B} + A \cdot B + A \cdot B + \bar{A} \cdot B &= A (\bar{B} + B) + B (A + \bar{A}) \\ &= A [1] + B [1] = A + B. \end{aligned}$$

We have also used relationship 3, but here we used $\bar{B} + B = 1$.

Implementing the Switching Functions

Implementing a Boolean expression in logic form requires an association between the basic logic functions and the logical connectives of the expression. The function:

$$\bar{C} \cdot (A + B) = Z,$$

for example, requires an AND gate with two inputs: $\bar{C}$ and $(A + B)$, as shown in Figure 29A. With C as a primary input, $\bar{C}$ is obtained by using an inverter (NOT-function). The NOT circuit is added in Figure 29B. The other input to the AND gate must be formed by a two-input OR gate which produces $(A + B)$. The complete logic arrangement is shown in Figure 29C.

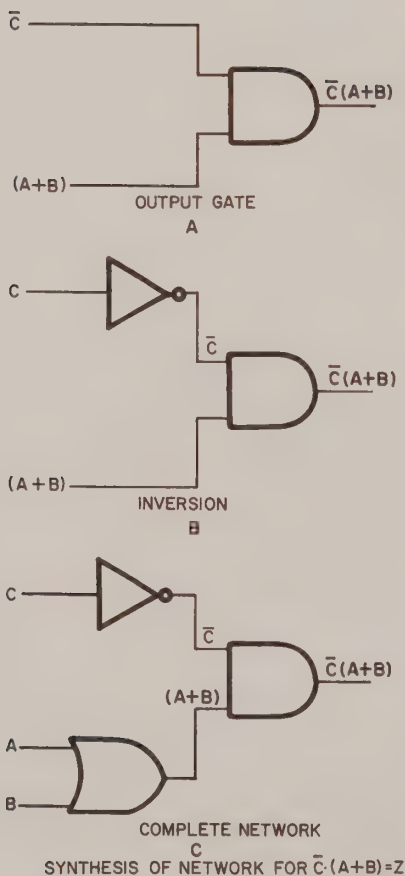


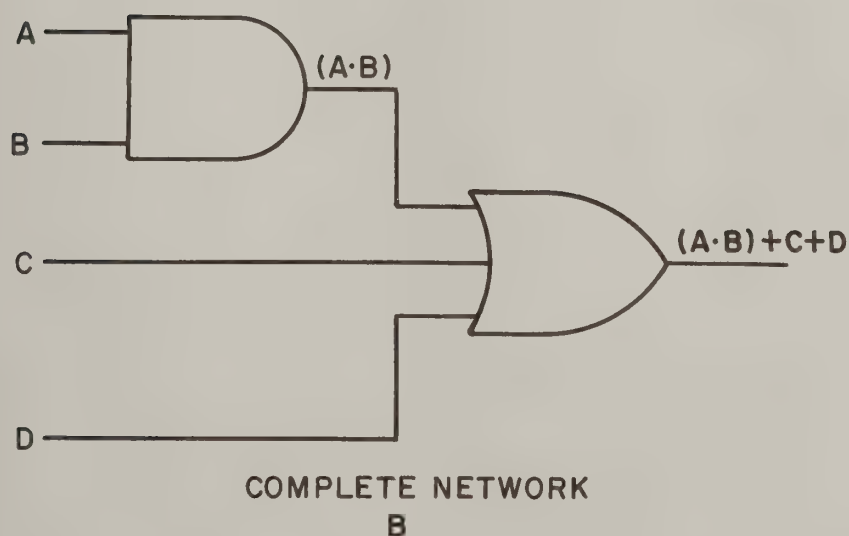
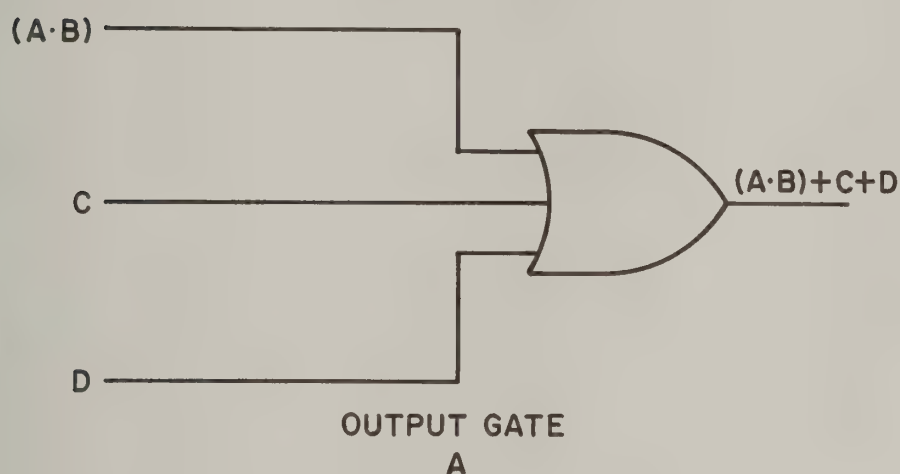
Figure 29

The function:

$$(A \cdot B) + C + D = Z$$

requires a three-input OR gate, as shown in Figure 30A. The input $(A \cdot B)$ requires an AND gate with two inputs. The complete network is shown in Figure 30B. Of course, there may be more than one possible implementation for a given expression. The example function may be thought of with $C + D$ grouped as a separate operation:

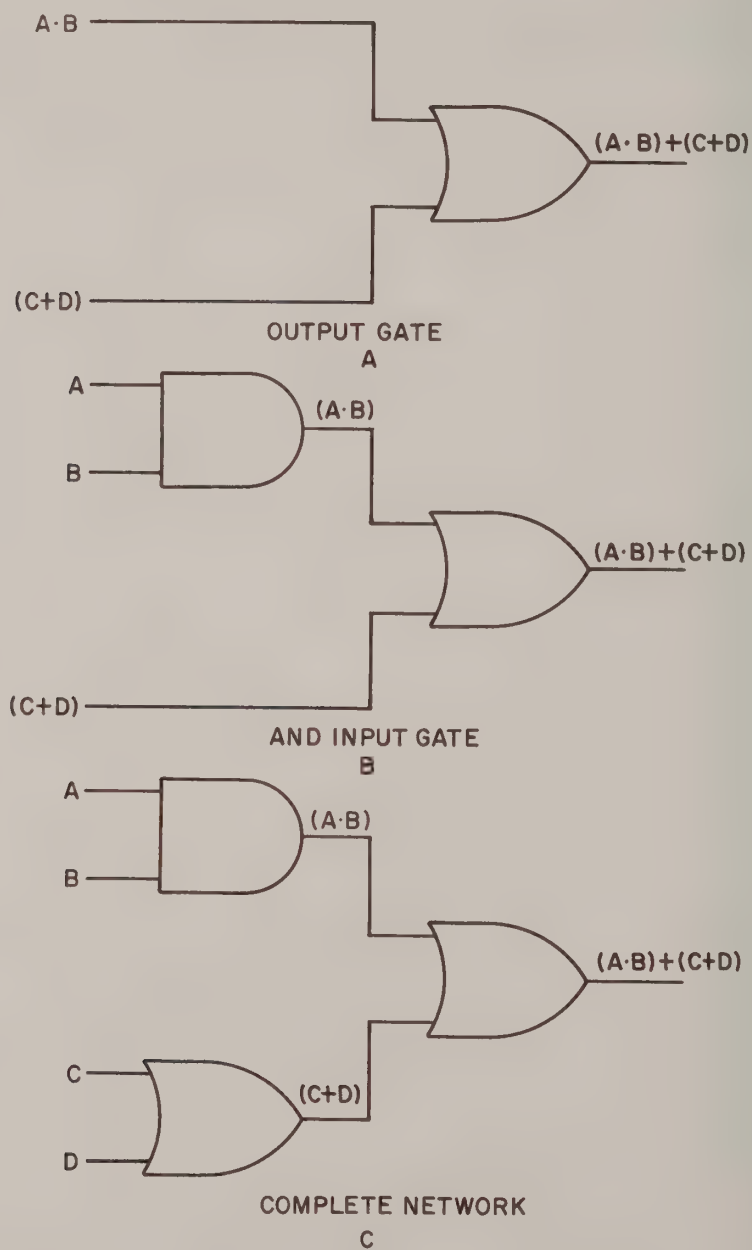
$$(A \cdot B) + (C + D) = Z.$$



SYNTHESIS OF NETWORK FOR $(A \cdot B) + C + D = Z$

Figure 30

Now, a two-input OR gate may be used with inputs $(A \cdot B)$ and $(C + D)$, as shown in Figure 31A. The first input requires an AND gate (Figure 31B); the second, another OR gate (Figure 31C). The implementation of the logic network shown in Figure 31 requires three components; that of Figure 30, only two components. Availability and cost of components determine the approach used.



SYNTHESIS OF NETWORK FOR $(A \cdot B) + (C + D) = Z$

Figure 31

Before implementing a logic network, it is wise to attempt every possible simplification of the Boolean expressions. The sum of products expression for the conditions in Table 19 has been shown to be:

$$(A \cdot B \cdot C) + (A \cdot \bar{B} \cdot C) + (\bar{A} \cdot B \cdot C) = Z.$$

Since C is common to all terms, it may be “factored out”;

$$C \cdot [(A \cdot B) + (A \cdot \bar{B}) + (\bar{A} \cdot B)] = Z.$$

The condition A is common to the first two terms inside the bracket. Hence, this too may be factored:

$$C \cdot [A(B + \bar{B}) + (\bar{A} \cdot B)] = Z.$$

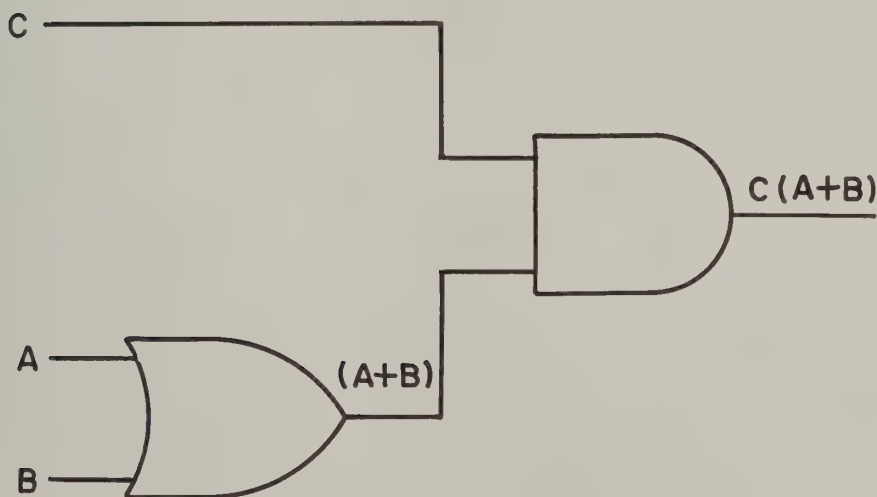
From Tables 26 and 27, $(B + \bar{B})$ is equal to 1 and $A \cdot 1$ is equal to A . This leads to:

$$C \cdot [A + (\bar{A} \cdot B)] = Z.$$

From Table 26, $A + (\bar{A} \cdot B)$ is the same as $(A + B)$;

$$C \cdot (A + B) = Z.$$

This final expression is implemented by the logic shown in Figure 32.



SYNTHESIS OF NETWORK FOR $C \cdot (A + B) = Z$

Figure 32

The sum of products form for Z_1 in Table 24 is:

$$\begin{aligned} &(\bar{A} \cdot \bar{B} \cdot C \cdot D) + (\bar{A} \cdot \bar{B} \cdot \bar{C} \cdot D) + (\bar{A} \cdot \bar{B} \cdot C \cdot \bar{D}) \\ &+ (\bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \bar{D}) + (A \cdot B \cdot \bar{C} \cdot \bar{D}) + (A \cdot \bar{B} \cdot \bar{C} \cdot \bar{D}) \\ &+ (\bar{A} \cdot B \cdot \bar{C} \cdot \bar{D}) = Z_1. \end{aligned}$$

Note that the fourth term, $(\bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \bar{D})$, has $\bar{A} \cdot \bar{B}$ in common with the first three terms and $\bar{C} \cdot \bar{D}$ in common with the last three terms. Therefore, $(\bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \bar{D})$ can be used twice:

$$\begin{aligned} &(\bar{A} \cdot \bar{B} \cdot C \cdot D) + (\bar{A} \cdot \bar{B} \cdot \bar{C} \cdot D) + (\bar{A} \cdot \bar{B} \cdot C \cdot \bar{D}) + (\bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \bar{D}) \\ &+ (\bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \bar{D}) + (A \cdot B \cdot \bar{C} \cdot \bar{D}) + (A \cdot \bar{B} \cdot \bar{C} \cdot \bar{D}) + (\bar{A} \cdot B \cdot \bar{C} \cdot \bar{D}) = Z_1. \end{aligned}$$

Then taking $\bar{A} \cdot \bar{B}$ out of the first four terms and $\bar{C} \cdot \bar{D}$ out of the rest:

$$\begin{aligned} &\bar{A} \cdot \bar{B} [C \cdot D + \bar{C} \cdot D + C \cdot \bar{D} + \bar{C} \cdot \bar{D}] \\ &+ \bar{C} \cdot \bar{D} [\bar{A} \cdot \bar{B} + A \cdot \bar{B} + \bar{A} \cdot B + A \cdot B] = Z_1. \end{aligned}$$

The expressions inside the brackets are both equal to 1 according to relationship 20 of Table 26.

$$\bar{A} \cdot \bar{B} [1] + \bar{C} \cdot \bar{D} [1] = \bar{A} \cdot \bar{B} + \bar{C} \cdot \bar{D} = Z_1.$$

This is a simplification of an equation having four variables in seven terms. Not all complicated equations will reduce this much. Some will not reduce at all. Two alternate implementations of this expression are shown in Figure 33.

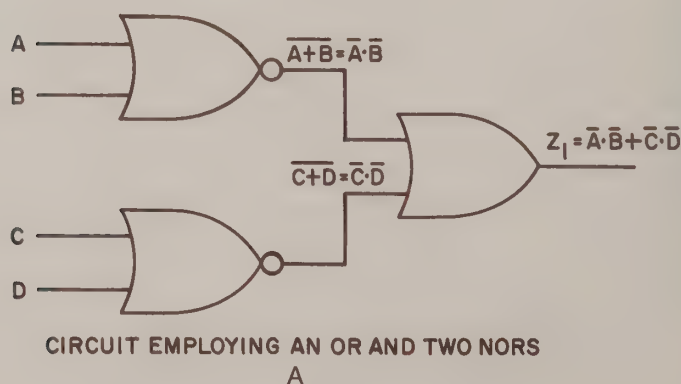
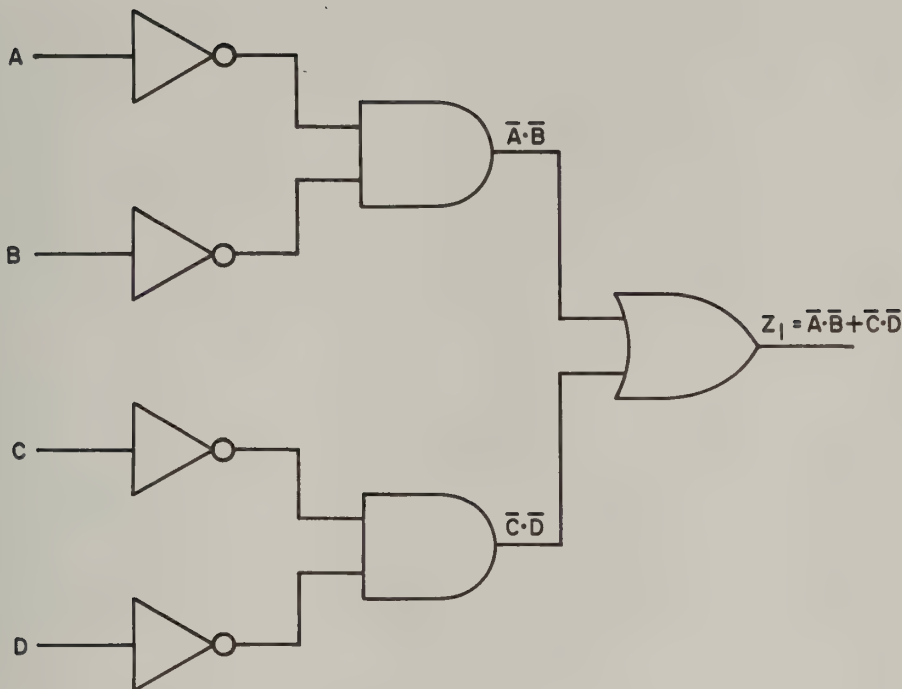


Figure 33



CIRCUIT EMPLOYING INVERTERS, ORS, AND ANDS
TWO IMPLEMENTATIONS OF $\bar{A} \cdot \bar{B} + \bar{C} \cdot \bar{D} = Z_1$

B

Figure 33

SUMMARY

Multilevel logic arrangements consist of any number of parallel or series combinations of the AND, OR, NOT, NAND or NOR functions. The EXCLUSIVE-OR function is a commonly used multilevel logic combination. The output for this function is a logical 1 only when either A or B is 1; but is logical 0 when A and B are 1 simultaneously, or when A and B are 0 simultaneously. Many different logic networks and associated switching functions may produce the EXCLUSIVE-OR and all other multilevel operations. However, when multiple networks or functions provide the same operation, their truth tables will always be identical.

Timing diagrams, switching functions and truth tables are three different methods for relating the outputs and inputs for logical networks. Each has its own advantages and disadvantages. The timing diagrams and truth tables are especially useful in analyzing and troubleshooting complex logic networks on a level-by-level basis. The timing diagrams are most appropriate for dynamic logic, which has rapidly changing inputs and outputs as a function of time. For such systems, the truth tables apply at each given timing interval; but a timing diagram or digital sequence must be used to "record" the values over a period of several timing intervals.

Switching functions are especially useful when it is necessary to design a network to produce a given operation. The operation may be originally specified as verbal statements, truth tables or switching functions. Given a verbal statement, a truth table can be developed. Given a truth table, a switching function can be developed. This is accomplished by interpreting the inputs as OR functions, where the outputs are logic 0's, then taking the complement of each digit and combining all the functions into a logical product. It can also be accomplished by interpreting the inputs where the outputs are logic 1's as AND functions, then combining all the functions into a logical sum.

Once a switching function has been associated with a desired logical operation, the first step (prior to implementing the logic network) is to simplify the function. This is accomplished by using basic laws of switching algebra; manipulating the switching function in a manner similar to rearranging and simplifying algebraic functions. (This approach can be used to derive a mathematical proof of the equivalence of two or more functions.) When a switching function cannot be simplified further, it can be implemented with a logic network having the fewest possible components.

The following Practice Exercise questions cover the subjects which you have just studied. They are:

DESIGNING COMBINATIONAL LOGIC

SIMPLIFICATION OF SWITCHING FUNCTIONS

IMPLEMENTING SWITCHING FUNCTIONS

40. Explain why $A + \bar{A} = Z$ means that Z is independent of the input condition.
41. Write $D(E + F)$ in an “expanded” form.
42. Write $C + (\bar{C} \cdot D)$ in a simplified form.
43. Explain why $(A \cdot B) \cdot [A + B + (\bar{A} \cdot C)]$ simplifies to $A \cdot B$.
44. Simplify: $A \cdot (A + B) \cdot (A + C) = Z$.
45. Simplify: $(A \cdot B) + (C \cdot D) + (A \cdot D) + A = Z$.
46. Explain how the logic in Figure 29 is derived for $\bar{C} \cdot (A + B) = Z$.
47. The logic arrangements in Figures 30 and 31 both implement the expression $(A \cdot B) + C + D = Z$. Why is the logic of Figure 30 preferred?
48. Implement a logic arrangement for Z_2 in Table 25 starting with the sum of products. Draw the circuits using (a) inverters, OR's and AND's and (b) using NOR's, NAND's, OR's and AND's as needed.
49. Implement a logic arrangement for the expression

$$(\bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \bar{D}) + (A \cdot \bar{B} \cdot \bar{C} \cdot \bar{D}) + (\bar{A} \cdot \bar{B} \cdot C \cdot D) = Z.$$

50. Prove that $A + (\bar{A} \cdot \bar{B}) = A + \bar{B}$.

51. Prove relationship 19 in Table 26.

APPENDIX

SWITCHING FUNCTION RELATIONSHIPS AND EQUIVALENCES

PART I

A good way to become familiar with the basic laws and relationships of switching algebra is to verify the relationships of Tables 26 and 27 by setting the variables equal to 0 and 1 in all possible combinations. This part of the Appendix shows how some of these relationships are verified in this way. In Table 27, it is stated that $A + 1 = 1$. Let $A = 0$. Then $0 + 1 = 1$, which checks. If $A = 1$, then $1 + 1 = 1$, which also checks. Thus, whether $A = 0$, or $A = 1$, the relationship $A + 1 = 1$ is true.

The first relationship in Table 26 states that $A + A = A$. If $A = 0$, then $0 + 0 = 0$, which checks. If $A = 1$, then $1 + 1 = 1$, which also checks. Thus, whether $A = 0$ or $A = 1$, the relationship $A + A = A$ is true.

Now check relationship 3 of Table 26, $A + \bar{A} = 1$. If $A = 0$, then $\bar{A} = 1$ and $0 + 1 = 1$, which checks. If $A = 1$, then $\bar{A} = 0$ and $1 + 0 = 1$, which also checks. Therefore, whether $A = 0$ or $A = 1$, the relationship $A + \bar{A} = 1$ is true. Relationship 9, $\bar{\bar{A}} = A$, is easy to check. If $A = 0$, then $\bar{A} = 1$; since $\bar{1} = 0$, $\bar{\bar{A}} = 0$. Also, if $A = 1$, then $\bar{A} = 0$; since $\bar{0} = 1$, $\bar{\bar{A}} = 1$. Therefore, whether $A = 0$ or $A = 1$, the relationship $\bar{\bar{A}} = A$ is true.

Now let's check relationship 15 of Table 26, $\bar{A} + AB = \bar{A} + B$, in this way. There are two binary variables in this relationship. Therefore, there are 2^2 or 4 possible pairs of values for these two variables. These 4 pairs of values are $A = 0$ and $B = 0$, $A = 0$ and $B = 1$, $A = 1$ and $B = 0$, and $A = 1$ and $B = 1$. Each of these possibilities must be checked. First, if $A = 0$ and $B = 0$, then $\bar{A} = 1$. Substituting these values in the equation gives $1 + 0 \cdot 0 = 1 + 0$. Now $0 \cdot 0 = 0$ and $1 + 0 = 1$. Therefore, $1 = 1$ and the relationship checks. Second, if $A = 0$ and $B = 1$, then $\bar{A} = 1$. Substituting these values in the equation gives $1 + 0 \cdot 1 = 1 + 1$. Now $0 \cdot 1 = 0$, $1 + 0 = 1$, and $1 + 1 = 1$. Therefore, $1 = 1$ and the relationship checks. Third, if $A = 1$ and $B = 0$, then $\bar{A} = 0$. Substituting these values in the equation gives $0 + 1 \cdot 0 = 0 + 0$. Now $1 \cdot 0 = 0$ and $0 + 0 = 0$. Therefore, $0 = 0$ and the relationship checks. And fourth, if $A = 1$ and $B = 1$, then $\bar{A} = 0$ and substituting these values in the equation gives $0 + 1 \cdot 1 = 0 + 1$. Now $1 \cdot 1 = 1$ and $0 + 1 = 1$. Therefore, $1 = 1$ and the relationship checks for all values of A and B in all possible combinations.

There is another way to establish relationship 15. Expand the equation:

$$\bar{A} + (A \cdot B) = \bar{A} \cdot (B + \bar{B}) + (A \cdot B) = \bar{A} \cdot B + \bar{A} \cdot \bar{B} + A \cdot B.$$

Note that $\overline{A} \cdot B$ has $\overline{A}$ in common with $\overline{A} \cdot \overline{B}$ and has B in common with $A \cdot B$. Boolean algebra allows us to use terms twice in the equations; thus

$$\overline{A} \cdot B + \overline{A} \cdot \overline{B} + \overline{A} \cdot B + A \cdot B =$$

$$\overline{A} (B + \overline{B}) + B (\overline{A} + A) =$$

$$\overline{A} [1] + B [1] = \overline{A} + B$$

where $\overline{A} \cdot B$ is used twice.

PART II

DeMorgan's Theorem expresses a basic duality, or two-foldness, which underlies all of switching algebra. All of the basic relationships of switching algebra can be divided into pairs in which one relationship is the dual, or twin, of the other. The dual of a relationship can be found by changing the signs in the relationship and by changing all 1's to 0's and all 0's to 1's. For example, $A + 0 = A$ is the dual of $A \cdot 1 = A$, and $A + (B + C) = (A + B) + C$ is the dual of $A(BC) = (AB)C$.

Notice that Table 26 is arranged to show the duality of the basic switching algebraic relationships. For the first eight relationships, the relationships at the right of the chart are the duals of those at the left. Then, relationships 10 and 11, 12 and 13, 14 and 15, and 16 and 17 are duals of each other. There is no dual for relationship 9 since it does not involve AND or OR symbols. In verifying and deriving these basic relationships, it is necessary to verify or derive only one relationship. After verifying or deriving one relationship, its dual must also be true. For example, if you verify the relationship $A + AB = A$, you have also verified the relationship $A(A + B) = A$, since the relationships are duals.

DeMorgan's Theorem can be expressed in such words as "break the bar and change the signs, or join the bars and change the signs." For example, in the expression $\overline{A + B}$, if you break the bar and change the sign, you have $\overline{A} \cdot \overline{B}$. Conversely, in the expression $\overline{A} \cdot \overline{B}$, if you join the bars and change the sign, you have $\overline{A + B}$. Hence, by DeMorgan's Theorem:

$$\overline{A + B} = \overline{A} \cdot \overline{B}$$

or vice versa. Similarly, in the expression $\overline{\overline{A} \cdot \overline{B}}$, if you break the bar and change the sign, you have $\overline{\overline{A} + \overline{B}}$. Or, in the expression $\overline{\overline{A} + \overline{B}}$, if you join the bar and change the sign, you have $\overline{\overline{A} \cdot \overline{B}}$. Hence:

$$\overline{\overline{A} \cdot \overline{B}} = \overline{\overline{A} + \overline{B}}$$

and vice versa. Notice that this method of expressing DeMorgan's Theorem only applies to expressions which have bars (negations or complementations).

In the text of this lesson, it is shown that the expressions (Equations 10 and 13, respectively):

$$(A + B) \overline{AB} = Z$$

$$\overline{\overline{(A + B)} + AB} = Z$$

both indicate the EXCLUSIVE-OR function, with Equation 10 being the fundamental form. This was proven by constructing level-by-level truth tables for Equation 13—showing the overall output-input values to be the same as those for Equation 10. We can also prove the equivalence mathematically by using DeMorgan's Theorem:

$$\begin{aligned}\overline{\overline{(A + B)} + AB} &= \overline{\overline{(A + B)}} \cdot \overline{AB} \\ &= (A + B) \overline{AB}.\end{aligned}$$

The first step breaks the top bar and changes the sign. The second step uses relationship 9 of Table 26.

It is also shown in the text of this lesson (Equation 12) that:

$$A \cdot \overline{B} + \overline{A} \cdot B = Z$$

is the EXCLUSIVE-OR function. By starting with Equation 10 and using DeMorgan's Theorem:

$$\begin{aligned}(A + B) \overline{AB} &= (A + B) (\overline{A} + \overline{B}) \\ &= A \cdot \overline{A} + A \cdot \overline{B} + B \cdot \overline{A} + B \cdot \overline{B} \\ &= A \cdot \overline{B} + B \cdot \overline{A} \\ &= A \cdot \overline{B} + \overline{A} \cdot B.\end{aligned}$$

The first step breaks the bar and changes the sign for the second factor. The second step expands the logical product on a term-by-term basis. The rest of the steps use relationships from Table 26.

Using DeMorgan's Theorem, the complement of any binary expression can also be obtained by changing the sign and placing a bar over each variable. Since any binary expression is equal to NOT its complement,

such as $A = \overline{\overline{A}}$, DeMorgan's Theorem may be expressed as "any binary expression is equal to NOT the expression obtained when the signs are changed and a bar is placed over each variable." For example, $A + B = \overline{\overline{A} \cdot \overline{B}}$, $\overline{A \cdot B} = \overline{\overline{A} + \overline{B}} = A + B$, $\overline{A + B} = \overline{\overline{A} \cdot \overline{B}} = \overline{A} \cdot \overline{B}$, and $AB = \overline{\overline{A} + \overline{B}}$. Notice that this method of expressing DeMorgan's Theorem applies to binary expressions which do not already have bars over the variables as well as those which do.

Many of the other equalities, proven in the lesson by truth table comparison, can be proven mathematically by using DeMorgan's Theorem and the relationships in Tables 26 and 27. For additional practice, you can go through several of these, such as Equations 5 through 8, Practice Exercise 9 and Equation 14.

IMPORTANT DEFINITIONS

COMBINATIONAL LOGIC—A multilevel logic network composed of various arrangements of AND, OR, NOT, NAND or NOR circuits.

DE MORGAN'S THEOREM—A theorem of SWITCHING ALGEBRA which expresses the principles of complementation and dualization in logic circuitry. DeMorgan's Theorem can be expressed in such words as "break the bar and change the signs or join the bars and change the signs." DeMorgan's Theorem can also be expressed as "any binary expression is equal to NOT the expression obtained when the signs are changed and a bar is placed over each variable." Other names for DeMorgan's Theorem are the involution laws, the inversion postulates and the dualization laws.

EXCLUSIVE-NOR—The logical function which allows an output only when all input conditions are simultaneously present or absent. It is the complementation of EXCLUSIVE-OR.

EXCLUSIVE-OR—The logical function which allows no output when all input conditions are simultaneously present or absent.

EXPANDED FORM—A switching function which has all of its LOGICAL PRODUCT components containing parenthetical LOGICAL SUMS written out completely on a term-by-term basis.

FACTOR—A component in a LOGICAL PRODUCT or AND expression.

INCLUSIVE-OR—The logical function which allows an output any time there is an input. Usually referred to simply as OR.

LOGICAL ADDITION—A term applied to the logical OR function. Also known as LOGICAL SUM.

LOGICAL MULTIPLICATION—A term applied to the logical AND function. Also known as LOGICAL PRODUCT.

LOGICAL PRODUCT—LOGICAL MULTIPLICATION.

LOGICAL SUM—LOGICAL ADDITION.

MINIMUM FORM—The switching function and associated logic network which accomplishes a given operation with the smallest number of interconnected basic logic units.

PRODUCT OF SUMS—The switching function obtained from a truth table by representing each row where the output is a logic zero with a LOGICAL SUM of the complements of each variable and taking the LOGI-

IMPORTANT DEFINITIONS (Continued)

CAL PRODUCTS of these rows. Also known as the OR-AND interpretation. Has been called the MAXTERM form.

SUM OF PRODUCTS—The switching function obtained from a truth table by taking the LOGICAL PRODUCT of the terms in each row where the output is a logical one and then taking the LOGICAL SUM of these products. Also known as the AND-OR interpretation. Has been called the MINTERM form.

SWITCHING ALGEBRA—The mathematics of logical circuitry. Originally developed for symbolic logic applications by George Boole. Also known as Boolean algebra.

TERMS—Components in a LOGICAL SUM or OR expression.

TIMING DIAGRAM—The set of electronic waveforms representing input and output conditions along a common time scale.

TIMING INTERVAL—The time between pulses in a digital input or output waveform. Chosen as a fundamental time unit for a digital system, all events are based on multiples of this interval.

ESSENTIAL SYMBOLS AND EQUATIONS

A, B, C, D	Arbitrarily selected input variables for switching networks
$\bar{A}, \bar{B}, \bar{C}, \bar{D}$	Inversion, negation or complementation of input variables
Z	Arbitrarily selected output variable for switching networks
$\bar{Z}$	Inversion, negation or complementation of output variable
$+$	Connective for OR (logical addition or logical sum)
$\cdot$	Connective for AND (logical multiplication or logical product)

$$A + B = Z \quad (\text{OR; INCLUSIVE-OR}) \quad (1)$$

$$A \cdot B = Z \quad (\text{AND}) \quad (2)$$

$$\overline{A + B} = Z \quad (\text{NOR}) \quad (3)$$

$$\bar{\bar{A}} = A \quad (\text{DOUBLE NEGATION}) \quad (4)$$

$$\overline{A + B} = A \cdot B \quad (5)$$

$$\overline{(A \cdot B)} = A + B \quad (6)$$

$$\overline{\overline{(A + B)}} = \overline{(A \cdot B)} \quad (7)$$

$$\overline{\overline{(A \cdot B)}} = \overline{(A + B)} \quad (8)$$

$$(A + B) \cdot C = (A \cdot C) + (B \cdot C) \quad (\text{EXPANDED FORM}) \quad (9)$$

$$(A + B) \overline{A \cdot B} = Z \quad (\text{EXCLUSIVE-OR}) \quad (10)$$

$$C \cdot \bar{D} = Z \quad (\text{INHIBIT}) \quad (11)$$

$$(A \cdot \bar{B}) + (\bar{A} \cdot B) = Z \quad (\text{EXCLUSIVE-OR}) \quad (12)$$

$$\overline{(A + B)} + \overline{(A \cdot B)} = (A + B) \cdot \overline{(A \cdot B)} = Z \quad (\text{EXCLUSIVE-OR}) \quad (13)$$

$$\overline{(A \cdot B)} + \overline{(A + B)} = \bar{A} \cdot \bar{B} + A \cdot B = Z \quad (\text{EXCLUSIVE-NOR}) \quad (14)$$

Also see Tables 26 and 27.

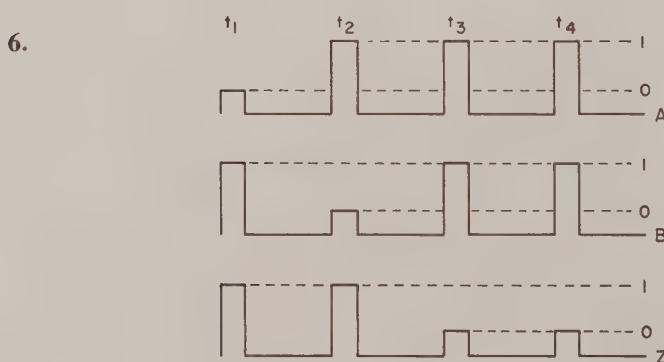
PRACTICE EXERCISE SOLUTIONS

1. True

2. Prior to the instant that A and B both go to logical 1, they are both logical 0. Hence, the output is controlled by $A + B = Z$ (or $A \cdot B = Z$). Two 0's applied to an OR gate produce a 0 output.

3. The truth table summarizes all possible output conditions for all possible input conditions. At each interval in the timing diagram, the instantaneous input conditions can be "looked up" in the truth table in order to determine the proper output condition.

4. Each input consists of a series of pulses, digits or conditions rather than a single condition.



7. True



PRACTICE EXERCISE SOLUTIONS (Continued)

9. The level-by-level development of the outputs for $\overline{\overline{A + B}}$ is shown in the table below. The truth table for $\overline{\overline{A + B}}$ consists of the first- and second-level input columns and the final output column. Comparing this to the truth table for $(A \cdot B)$, which is the NAND function (Table 4 of the text), you can see that the two functions are the negation of each other,

$$\overline{\overline{A + B}} = \overline{Z}.$$

Thus, negating each side produces

$$\overline{\overline{\overline{\overline{A + B}}}} = \overline{\overline{Z}} = Z = \overline{\overline{(A \cdot B)}}.$$

First-Level Inputs		Second-Level Inputs		Second-Level Output	Final Output
A*	B*	$\overline{A}$	$\overline{B}$	$\overline{\overline{A + B}}$	$\overline{\overline{A \cdot B}} = Z^*$
0	0	1	1	0	1
1	0	0	1	0	1
0	1	1	0	0	1
1	1	0	0	1	0

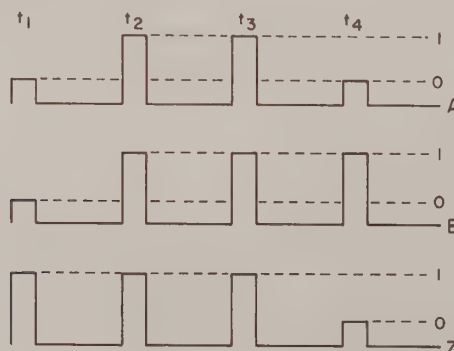
* Compare to Table 4

10. The inputs to the first level are A and B. The first-level function is OR, resulting in an output of $(A + B)$. This output, in conjunction with C, forms the input to the second level. The second-level function is AND, leading to the output $(A + B) \cdot C$.
11. With A and B inputs as 1 and 0 (respectively) to the first-level OR gate, the first-level output is a logical 1. This logical 1 and C, which is also 1, are inputs to the second-level AND gate, resulting in a 1 output.
12. The second-level AND gate produces a 0 output when either or both of its inputs are 0. Since C is 0 in the last four rows of Table 9, all of the second-level outputs must be 0, regardless of the values of A and B.
13. False
14. True
15. 0010.
16. The first level is composed of two AND gates. The output of one AND gate is $A \cdot B$; of the other, $C \cdot D$. With these as inputs, the second-level OR gate produces $(A \cdot B) + (C \cdot D)$. The third-level NOT gate inverts this to $\overline{(A \cdot B) + (C \cdot D)}$.

PRACTICE EXERCISE SOLUTIONS (Continued)

17. Using the AND-NOR logic of Figure 16, with 0 and 1 ($\bar{A}$ and B) applied to one AND gate in the first level, the output will be 0. With two 0's (C and D) applied to the second AND gate in the first level, the output will also be 0. The NOR gate in the second level produces a 1 output when both inputs are 0.
18. 1010.
19. The three outputs from the first level are $(A \cdot B \cdot C)$, $(C \cdot D)$ and $(D + E)$. The first two of these form the inputs for the second-level OR gate, resulting in a second-level output of $(A \cdot B \cdot C) + (C \cdot D)$. This, in conjunction with $(D + E)$, forms the input to the third-level AND gate.
20. $(E + F) \overline{(E \cdot F)} = Z$.
21. False
22. With a 0 and 1 input, the OR gate in the first level produces a 1 output which is applied to the positive-logic terminal of the INHIBIT gate. With a 0 and 1 input on the AND gate in the first level, the output is 0 and, since this is applied to the INHIBIT terminal of the second-level gate, there is no inhibiting action. Hence, the final output is 1.
23. True
24. Since the inputs are both 1, both first-level INHIBIT gates are inhibited, resulting in two 0's being applied to the second-level OR gate. This produces a 0 at the final output.
25. The output-input relationships, as summarized by the truth table of a logic network.
26. When A is 1 and B is 0, the first-level NOR gate produces a 0 output; the first-level AND gate, a 0 output. With these as inputs, the second-level NOR gate produces a 1 output.

27.



PRACTICE EXERCISE SOLUTIONS (Continued)

28. A “plain English” statement of conditions and results, a truth table or a Boolean expression (switching function).
29. By the first statement of conditions and results, Z can be 1 only when C is 1. Conversely, if C is 0, Z must be 0.
30. By the second statement of conditions and results, when C and D are both 0 and A and B are both 1, Z₂ must be 0.

31.

A	B	C	Z
1	1	1	1
1	0	1	1
0	1	1	1
0	0	1	1
1	1	0	1
1	0	0	0
0	1	0	0
0	0	0	0

32. True

33. One approach assumes that the input conditions which result in 0 outputs represent OR functions and replaces these input digits with complementary Boolean symbols; the second approach assumes that the input conditions which result in 1 represent AND functions, again replacing the input digits with Boolean symbols.
34. Product of sums expressions use the OR-interpretation of the complement of the individual rows of a truth table which result in 0 outputs and join them as AND terms to form an OR-AND expression.
35. Sum of products expressions use the AND-interpretation of the individual rows of a truth table which result in 1 outputs and join them as OR components to form an AND-OR expression.

36. True

$$37. (A \cdot B \cdot C) + (A \cdot \bar{B} \cdot C) + (\bar{A} \cdot B \cdot C) = Z.$$

$$38. (\bar{A} + \bar{B} + \bar{C} + \bar{D}) \cdot (\bar{A} + B + \bar{C} + \bar{D}) \cdot (A + \bar{B} + \bar{C} + \bar{D}) \\ \cdot (\bar{A} + \bar{B} + C + \bar{D}) \cdot (\bar{A} + B + C + \bar{D}) \cdot (A + \bar{B} + C + \bar{D}) \\ \cdot (\bar{A} + \bar{B} + \bar{C} + D) \cdot (\bar{A} + B + \bar{C} + D) \cdot (A + \bar{B} + \bar{C} + D) = Z_1.$$

PRACTICE EXERCISE SOLUTIONS (Continued)

$$39. (A + B + C) \cdot (\bar{A} + \bar{B} + C) \cdot (A + \bar{B} + \bar{C}) \cdot (\bar{A} + \bar{B} + \bar{C}) \cdot (A + \bar{B} + C) \\ = Z \text{ (Product of sums).}$$

$$(A \cdot \bar{B} \cdot \bar{C}) + (\bar{A} \cdot \bar{B} \cdot C) + (A \cdot \bar{B} \cdot C) = Z \text{ (Sum of products).}$$

40. If A is either a logical 1 or a logical 0, $(A + \bar{A})$ will produce a logical 1 output; the output is independent of the input condition.

$$41. D(E + F) = (D \cdot E) + (D \cdot F).$$

$$42. C + (\bar{C} \cdot D) = C + D.$$

43. The function can be expanded to:

$$(A \cdot B \cdot A) + (A \cdot B \cdot B) + (A \cdot B \cdot \bar{A} \cdot C).$$

However, $A \cdot A$ reduces to A, $B \cdot B$ reduces to B, and $A \cdot \bar{A}$ reduces to 0, leaving:

$$(A \cdot B) + (A \cdot B) + (0 \cdot B \cdot C).$$

The first two terms are redundant; the third term is 0, leaving only $A \cdot B$.

44. From relationship 8 of Table 26, $(A + B) \cdot (A + C)$ is the same as $A + (B \cdot C)$. Hence:

$$A \cdot (A + B) \cdot (A + C) = A \cdot [A + (B \cdot C)].$$

Expanding this yields:

$$(A \cdot A) + (A \cdot B \cdot C).$$

From relationship 2 of Table 26, $A \cdot A$ is the same as A, reducing the expression to:

$$A + (A \cdot B \cdot C).$$

Since A is common to both terms, the expression may be “factored” in an algebraic manner:

$$A \cdot [1 + (B \cdot C)].$$

From relationship 3 of Table 27, the bracketed expression $1 + (B \cdot C)$ is simply 1, leading to the final reduction:

$$A = Z.$$

PRACTICE EXERCISE SOLUTIONS (Continued)

45. From relationship 11 of Table 26, $(A \cdot B) + A$ is the same as A . Hence:

$$(A \cdot B) + (C \cdot D) + (A \cdot D) + A = A + (C \cdot D) + (A \cdot D).$$

The same relationship indicates that $A + (A \cdot D)$ is the same as A , leading to:

$$A + (C \cdot D) = Z$$

as the simplified expression.

46. $\bar{C} \cdot (A + B)$ requires an AND gate with two inputs: $\bar{C}$ and $(A + B)$. With C as a primary input, $\bar{C}$ is obtained with an inverter (NOT-function) preceding one of the AND-gate input terminals. The other input to the AND gate must be formed by a two-input OR gate with A and B as its primary inputs.

47. The arrangement in Figure 30 requires fewer components. It has two gates compared to three for Figure 31.

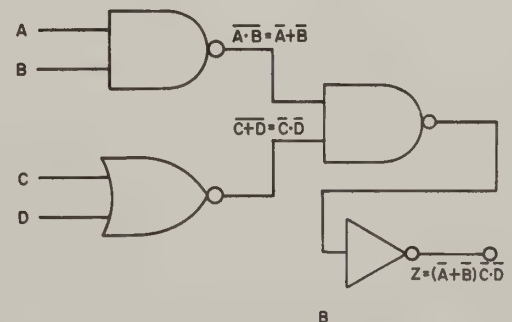
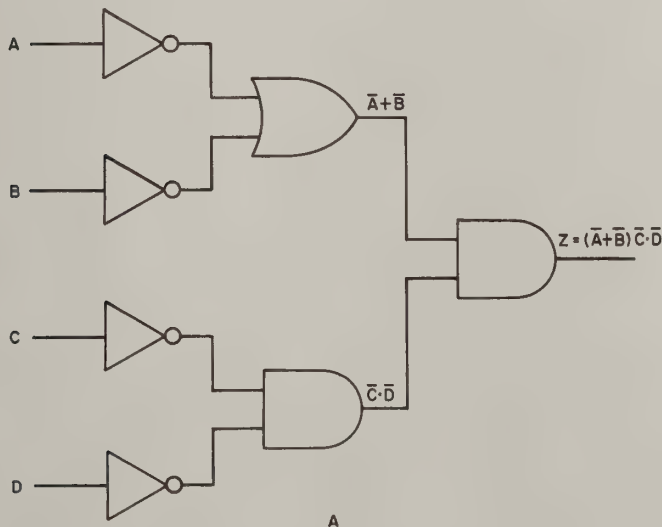
48. The expression for Z_2 is:

$$(A \cdot \bar{B} \cdot \bar{C} \cdot \bar{D}) + (\bar{A} \cdot B \cdot \bar{C} \cdot \bar{D}) + (\bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \bar{D}) = Z_2.$$

Since $\bar{C} \cdot \bar{D}$ are common to all three terms, they may be factored:

$$(\bar{C} \cdot \bar{D}) \cdot [(A \cdot \bar{B}) + (\bar{A} \cdot B) + (\bar{A} \cdot \bar{B})] = \bar{C} \cdot \bar{D} \cdot (\bar{A} + \bar{B}).$$

Using relationship 19 of Table 26, this can be implemented by the following logic:



PRACTICE EXERCISE SOLUTIONS (Continued)

49. First, the expression must be simplified. There are several ways to do this. However, since $\bar{B} \cdot \bar{C} \cdot \bar{D}$ is common to the first two terms, this logical product may be factored:

$$\bar{B} \cdot \bar{C} \cdot \bar{D} (\bar{A} + A) + \bar{A} \cdot \bar{B} \cdot C \cdot D = Z.$$

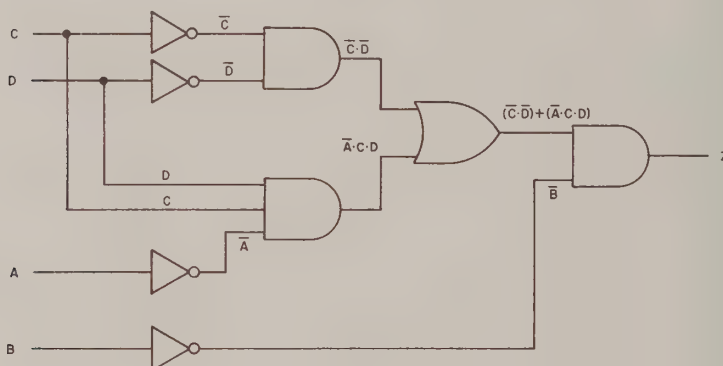
However, from Table 26: $\bar{A} + A = 1$. Hence:

$$\bar{B} \cdot \bar{C} \cdot \bar{D} + \bar{A} \cdot \bar{B} \cdot C \cdot D = Z.$$

Now $\bar{B}$ is common to both terms and may be factored:

$$\bar{B} [\bar{C} \cdot \bar{D} + \bar{A} \cdot C \cdot D] = Z.$$

The logic is implemented below.



50. Expanding gives:

$$A + (\bar{A} \cdot \bar{B}) = A (B + \bar{B}) + (\bar{A} \cdot \bar{B}) = A \cdot B + A \cdot \bar{B} + \bar{A} \cdot \bar{B}.$$

$A \cdot \bar{B}$ has A in common with $A \cdot B$ and has $\bar{B}$ in common with $\bar{A} \cdot \bar{B}$; thus, $A \cdot \bar{B}$ can be used twice.

$$A + (\bar{A} \cdot \bar{B}) = A \cdot B + A \cdot \bar{B} + A \cdot \bar{B} + \bar{A} \cdot \bar{B}$$

$$= A (B + \bar{B}) + \bar{B} (A + \bar{A})$$

$$= A [1] + \bar{B} [1] = A + \bar{B}.$$

51. In $A \cdot B + A \cdot \bar{B} + \bar{A} \cdot B$, $A \cdot B$ has A in common with $A \cdot \bar{B}$ and has B in common with $\bar{A} \cdot B$. Therefore, $A \cdot B$ can be used twice:

$$A \cdot B + A \cdot \bar{B} + \bar{A} \cdot B = A \cdot B + A \cdot \bar{B} + A \cdot B + \bar{A} \cdot B =$$

$$A \cdot (B + \bar{B}) + B \cdot (A + \bar{A}) = A \cdot [1] + B \cdot [1] = A + B.$$



ONE OF THE

BELL & HOWELL SCHOOLS

1. C - TRUTH TABLES, SWITCHING FUNCTIONS, AN TIMING DIAGRAMS -- PROVIDE DIFFERENT METHODS FOR EXPRESSING THE OUTPUT-INPUT RELATIONSHIPS FOR DIGITAL SYSTEMS.

Truth tables list, in tabular form, the inputs to a particular gate and the resultant output of that gate. Switching functions use Boolean algebra to express the various output-input switching relationships. Timing diagrams graphically illustrate the input waveforms and the resultant output waveform. All three methods can be used with either static or dynamic logic.

2. C - THE TRUTH TABLE FOR A FOUR INPUT LOGIC SYSTEM -- WILL HAVE SIXTEEN POSSIBLE INPUT COMBINATIONS.

The equation for determining the input combinations of any logic system is stated as 2^n , where n is the number of actual inputs. In this case we have a four input logic system; therefore, our equation is 2^4 which is 16. Thus, we will have sixteen possible input combinations.

3. B - IF AN INVERT (OR NOT) STAGE WERE APPLIED BETWEEN THE FIRST-LEVEL AND GATE FOR A AND B AND THE SECOND-LEVEL OR GATE IN FIGURE 17, THE OUTPUT WOULD BE -- $(\overline{A \cdot B}) + (C \cdot D)$.

The output at the first AND gate would be $A \cdot B$, the output of the second AND gate would be $C \cdot D$. Since the NOT stage is placed between the first-level AND gate, the input to the OR gate would be $\overline{A \cdot B}$ and $C \cdot D$. The output of the second-level OR gate would be $(\overline{A \cdot B}) + (C \cdot D)$. This output is then fed to another NOT stage, resulting in an output of $\overline{(\overline{A \cdot B}) + (C \cdot D)}$ at Z.

4. C - IF THE NOT STAGE AT THE END OF THE SYSTEM IN FIGURE 17 WERE REMOVED, THE OUTPUT COLUMN OF THE TRUTH DATA IN TABLE 11 -- WOULD HAVE ALL 1'S CHANGED TO 0'S AND ALL 0'S CHANGED TO 1'S.

The output at Z now becomes $(A \cdot B) + (C \cdot D)$. By substituting logical 1's and 0's for the inputs at A, B, C and D in accordance with Table 11, the output at Z will become 1 instead of 0, and 0 instead of 1.

5. C - $(A \cdot B) + (A \cdot \overline{B}) = Z$; $(A + B) \cdot (A \cdot \overline{B}) = Z$; and $(\overline{A + B}) + (A \cdot B) = Z$ ARE -- THREE ALTERNATE FORMS OF THE EXCLUSIVE-OR FUNCTION.

The EXCLUSIVE-OR FUNCTION PRODUCES AN OUTPUT (logic 1) only when either inputs are at 1 and it does not produce an output when both inputs are logical 1. These three equations functionally represent three different methods employed to obtain the EXCLUSIVE-OR function. $(A \cdot \overline{B}) + (\overline{A} \cdot B)$ employs an INHIBIT function at the B input to the first AND gate and the A input to the second AND gate. The two AND gates are then OR'ed together to obtain the EXCLUSIVE-OR function. $(A + B) \cdot (\overline{A \cdot B})$ employs an AND NOR gate feeding an OR gate to obtain the EXCLUSIVE-OR function. $(\overline{A + B}) + (A \cdot B) = Z$ employs a NOR gate and an OR gate feeding another NOR gate to obtain the EXCLUSIVE-OR function.

6. B - THE EXCLUSIVE-NOR FUNCTION -- IS THE COMPLEMENT OF THE EXCLUSIVE-OR FUNCTION.

The inversion or complementation of the EXCLUSIVE-OR function produces the EXCLUSIVE-NOR function. The EXCLUSIVE-NOR truth table may be obtained by complementing the outputs of the EXCLUSIVE-OR truth table. The complement of the INCLUSIVE-OR function is the INCLUSIVE-NOR function.

7. D - THE EXPRESSION; $(A \cdot \overline{B} \cdot \overline{C} \cdot \overline{D}) + (A \cdot B \cdot \overline{C} \cdot \overline{D}) + (\overline{A} \cdot \overline{B} \cdot C \cdot D)$ IS -- A SUM OF THE PRODUCTS EXPRESSION.

The OR-AND expression or product of the sums expression would be written as $(A + \overline{B} + \overline{C} + \overline{D}) \cdot (\overline{A} + B + \overline{C} + \overline{D}) \cdot (\overline{A} + \overline{B} + C + D)$. The sum of the products expression as stated is not in its simplest form.

8. A - THE EXPRESSION: $(A \cdot B \cdot C \cdot D) + (A \cdot B \cdot C \cdot \overline{D}) + (\overline{A} \cdot \overline{B} \cdot C \cdot D)$ SIMPLIFIES TO -- $C \cdot [(A \cdot B) + (\overline{A} \cdot \overline{B} \cdot D)]$.

First, factor out the C term. The expression then becomes: $C \cdot (A \cdot B \cdot D) + (A \cdot B \cdot \overline{D}) + (\overline{A} \cdot \overline{B} \cdot D)$. Since $D + \overline{D} = 1$, next, factor out $A \cdot B$. This results in the following expression: $C \cdot (A \cdot B) \cdot [D + (\overline{D})] + (\overline{A} \cdot \overline{B} \cdot D)$ expression further simplifies to $C \cdot [(A \cdot B) + (\overline{A} \cdot \overline{B} \cdot D)]$.



5225A (CON'T)

9. D - THE EXPRESSION: $(A + B) \cdot (A + C) \cdot (B + C)$ SIMPLIFIES TO -- $C \cdot (A + B) + (A \cdot B)$.

From the initial expression: $(A + B) \cdot (A + C) \cdot (B + C)$, the second and third terms: $(A + C) \cdot (B + C)$ may be simplified to $C + (A \cdot B)$. The entire expression then becomes:

$$(A + B) \cdot C + (A \cdot B)$$

Rearranging the product order of the first two terms yields:

$$C \cdot (A + B) + (A \cdot B)$$

10. D - THE EXPRESSION: $C \cdot [D + A \cdot B]$ CAN BE IMPLEMENTED WITH -- TWO 2-INPUT AND GATES AND A 2-INPUT OR GATE.

The input to the first gate would be A and B. The output from the first AND gate ($A \cdot B$) is used as one input to the OR gate, the other input being D. The output from the OR gate $[D + A \cdot B]$ is used as one input to the other AND gate. The other input to the second AND gate is C. The output of the second AND gate is $C \cdot [D + A \cdot B]$.

20,000

A	B	Z
0	0	0
0	1	1
1	0	1
1	1	0

QUESTIONS

IMPORTANT—These instructions **MUST** be accurately followed to avoid loss, or errors in grading.

Indicate your answer on this sheet by filling in the box for the most correct answer to each question, as illustrated in the example below.

When all questions have been answered, place the answer card in the proper position to line up the boxes on the card with the boxes on the sheet.

Next, copy the complete lesson code into the space provided on the card, and fill in the answer boxes to correspond with those previously filled in on this sheet.

Before mailing, be certain your correct student number, name and address appear on the card.

LESSON CODE
5225A

Example: United States Senators belong to which branch of national government?
 (A) Executive. (B) Legislative. (C) Judicial. (D) The Cabinet.

A	☐
B	☒
C	☐
D	☐

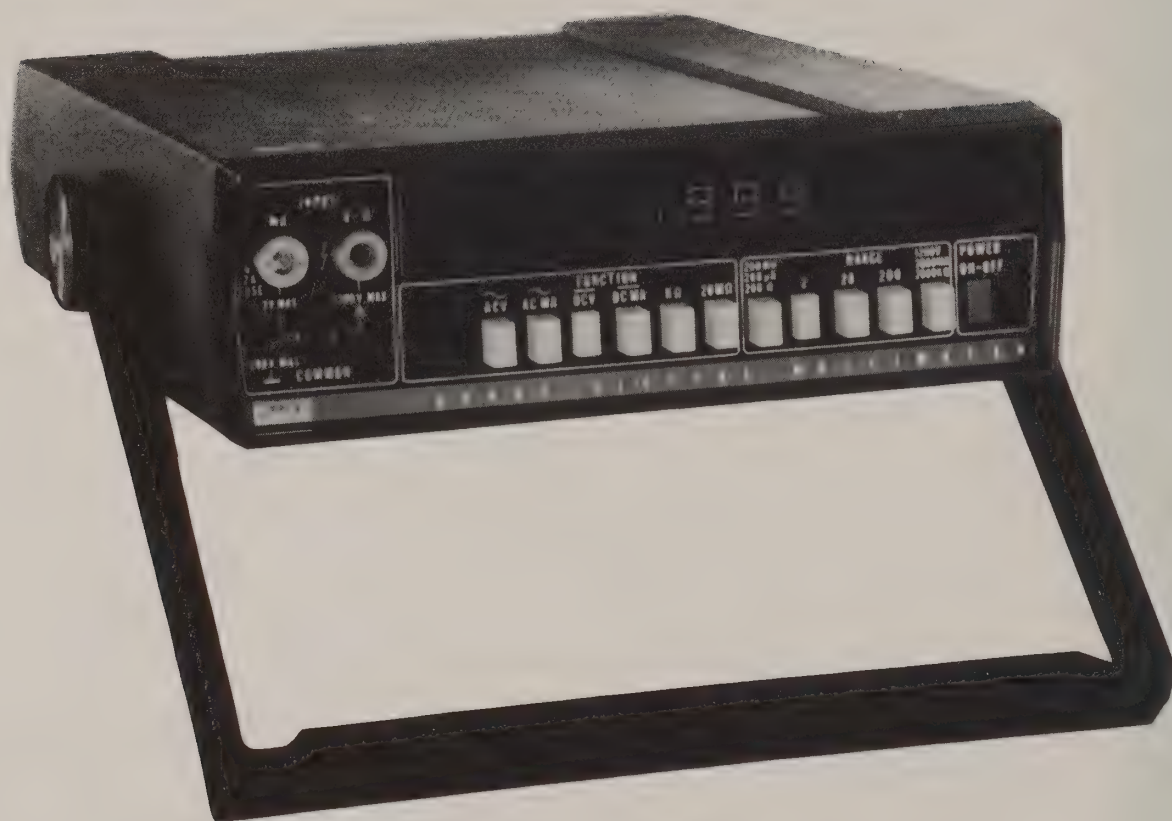
- Truth tables, switching functions and timing diagrams**
 (A) have nothing in common. (B) are used only for static logic systems. (C) provide different methods for expressing the output-input relationships for digital systems. (D) are used only for dynamic logic systems.
- The truth table for a four-input logic system**
 (A) will have four possible input combinations. (B) will have eight possible input combinations. (C) will have sixteen possible input combinations. (D) will have thirty-two possible output combinations.
- If an INVERT (or NOT) stage were applied between the first-level AND gate for A and B and the second-level OR gate in Figure 17, the output would be:**
 (A) $(A \cdot B) + (C \cdot D)$. (B) $(\overline{A \cdot B}) + (C \cdot D)$. (C) $(A + B) \cdot (C + D)$. (D) $\overline{A \cdot B \cdot C \cdot D}$.
- If the NOT stage at the end of the system in Figure 17 were removed, the output column of the truth data in Table 11**
 (A) would not be altered. (B) would be changed only when A and B are both 0. (C) would have all 1's changed to 0 and all 0's changed to 1's. (D) would be changed only when C and D are both 1.
- $(A \cdot B) + (\overline{A} \cdot \overline{B}) = Z$; $(A + B) \cdot (\overline{A} \cdot \overline{B}) = Z$ and $(A + B) + (\overline{A} \cdot \overline{B}) = Z$**
 (A) are three alternate forms of the EXCLUSIVE-NOR function. (B) are special combinational functions available only in semiconductor form. (C) are three alternate forms of the EXCLUSIVE-OR function. (D) are three multilevel switching functions with nothing in common.
- The EXCLUSIVE-NOR function**
 (A) is the complement of the INCLUSIVE-OR function. (B) is the complement of the EXCLUSIVE-OR function. (C) has the Boolean expression $\overline{A \cdot B} + A \cdot B$. (D) is the dual of the INCLUSIVE-OR function.
- The expression: $(A \cdot \overline{B} \cdot \overline{C} \cdot \overline{D}) + (\overline{A} \cdot B \cdot \overline{C} \cdot \overline{D}) + (\overline{A} \cdot \overline{B} \cdot C \cdot \overline{D})$ is**
 (A) an OR-AND expression. (B) a product of sums expression. (C) in its simplest possible form. (D) a sum of products expression.
- The expression: $(A \cdot B \cdot C \cdot D) + (A \cdot B \cdot C \cdot \overline{D}) + (\overline{A} \cdot \overline{B} \cdot C \cdot D)$ simplifies to**
 (A) $C \cdot [(A \cdot B) + (\overline{A} \cdot \overline{B} \cdot D)]$. (B) $(C \cdot D) + (A \cdot B)$. (C) $C \cdot [D + (\overline{A} \cdot B)]$. (D) $(A \cdot B) + (C \cdot D)$.
- The expression: $(A + B) \cdot (A + C) \cdot (B + C)$ simplifies to**
 (A) $A \cdot (B + C) + (A \cdot B)$. (B) $B \cdot (A + C) + (A \cdot B)$. (C) $C \cdot [(A + B) + (A \cdot B)]$. (D) $C \cdot (A + B) + (A \cdot B)$.
- The expression: $C \cdot [D + (A \cdot B)]$ can be implemented with**
 (A) a 3-input AND gate and a 2-input OR gate. (B) two 2-input OR gates and a 2-input AND gate. (C) a 2-input AND gate and a 2-input OR gate. (D) two 2-input AND gates and a 2-input OR gate.

CODING AND DECODING CIRCUITS

5230

© COPYRIGHT 1973
BELL & HOWELL SCHOOLS, INC.
ALL RIGHTS RESERVED





This digital multimeter converts the input signal (current, voltage or resistance) to dc voltage. This dc voltage is changed to an ac signal whose frequency is determined by the amplitude and polarity of the dc voltage. A digital IC counts the frequency. The count is transferred in Binary Coded Decimal (BCD) form to the display section. The display section decodes the BCD signal and drives the read out.

Courtesy John Fluke Mfg. Co., Inc.

CONTENTS

Number Systems	Page 4
Decimal Numbers	Page 6
Binary Numbers	Page 8
Binary-Coded Decimal Numbers	Page 13
The Excess-3 Code	Page 15
The Gray Code	Page 16
Decoding Logic	Page 19
AND Expression Decoding	Page 21
Pyramid Decoding	Page 26
Matrix Decoding	Page 31
Special Decoder Logic	Page 32
Encoding Logic	Page 36
Parity Checking Logic	Page 38
Computer Codes	Page 41
Alphanumeric Coding	Page 42
Specific Computer Codes	Page 43

*A stern discipline pervades all nature,
which is a little cruel that it may be
very kind.*

—Edmund Spenser

CODING AND DECODING CIRCUITS

A language is made up of characters such as the letters of the alphabet, numbers and other symbols. Intelligent people group letters together to form words. In turn, they use the words to convey information to other intelligent people.

It is also possible to convey information by means of digital systems. A human operator feeds information to such a system. In turn, the system processes the information. After the processing is complete, it may feed the results back to the human operator or use them to control another machine.

Although the information fed to and received from a digital system may be in a particular human language form, the system does not process it in this form. The system uses its own set of symbols to represent particular human language characters. This set of symbols is called a **CODE**.

In this lesson we will be primarily concerned with the use of coded numbers, rather than coded alphabetical characters. However, both types of information are represented by combinations of only two symbols (0 and 1). These combinations are known as the **BINARY CODE** to distinguish them from the multitude of other codes that could be developed. Many types of binary codes can also be formed. We will describe the ones most commonly used.

The **DECIMAL NUMBER SYSTEM** is the system we are most accustomed to working with. It is also the system we use to “input” data into and “output” data from digital devices. For example, we may enter several decimal numbers into a digital computer. The computer may perform a variety of mathematical operations on these numbers using some sort of a binary code. The computer may then print out or display the result in decimal form. A typical display for digital devices is shown in Figure 1. Since the “internal processing” is in binary form and the input and output are in decimal form, the device must contain circuitry which converts from one form to the other. We will describe typical **ENCODING CIRCUITS** and **DECODING CIRCUITS** used to accomplish these conversions.

NUMBER SYSTEMS

Before studying the various binary codes, it will be helpful to take a closer look at our own decimal system. A deeper understanding of decimal numbers makes it much easier to work with less familiar binary numbers. You probably have a fair understanding of decimal numbers already, but it is easy to take for granted those things with which you are familiar. You



*A typical decimal display for a digital system.
(Courtesy Non-Linear Systems, Inc.)*

Figure 1

automatically carry out a procedure long after some of the rules are forgotten.

The symbol for “none” or “nothing” is very basic to every number system. On paper, none or nothing is usually indicated by the Arabic symbol “0,” or zero. Every number system has some symbol or condition to indicate a single item. In Arabic numbers, this is the familiar “1.” Although other number systems use other symbols for 0 and 1, separate symbols are almost always used to designate “nothing” and “one.”

When it comes to designating some specific value greater than one, a number of possible systems can be utilized. In one extreme case, the same symbol used for 1 could be repeated the number of times necessary to indicate the value at hand. An example of this type of system is the tally used for inventories, elections, etc. It has the nice feature that the number can be increased simply by adding more ones. For example, a count of 12, indicated as 1111 1111 11, can be increased to 14 by simply adding two ones to give 1111 1111 1111. This number system requires the knowledge of only two symbols, 0 (still used to indicate “nothing”) and 1. A large number written in this way would be hard to read and would take up a lot of space. Such a system would be hard to use in solving problems, especially where large numbers are involved.

A second extreme is to devise and remember a different symbol for each number. Although such a system would make for a compact notation, a large number of symbols would be needed. For example, to work problems that involve quantities up to one million would require a million different symbols.

Decimal Numbers

A more practical approach is to use a limited number of symbols and devise a pattern in which these symbols should be arranged to indicate larger numbers. Although any number of symbols can be used in such a system, the popular one is based on the ten Arabic symbols: 0,1,2,3,4,5,6,7,8, and 9. Possibly this choice is due to the fact that prehistoric man counted on his fingers. The number of distinct symbols used is called the **RADIX** of the system. The decimal system has a radix of 10.

Since the decimal number system is based on a radix of 10, a review of the powers of 10 will be helpful to you in understanding the fine points of this number system. A number multiplied by itself results in a power of that number. For example, 100 is the second power of 10 because $10 \times 10 = 100$.

The power of a number is often expressed in exponential form. This means that a number (called an exponent) is written to the upper right of the ten to indicate how many of the tens are to be multiplied together. For example, $100 = 10 \times 10 = 10^2$. An exponent 2 is placed to the upper right of the 10 to show that two tens are to be multiplied together.

10^4	10^3	10^2	10^1	10^0	Power of Ten
10,000	1000.0	100.00	10.000	1.0000	Decimal Value

Digit Position Values for Whole Numbers (Positive Powers of 10)

Table 1

Other powers of ten are shown in Tables 1 and 2. Notice, in Table 1, that as the positive exponent decreases by one unit, the decimal point in the resulting number moves one place to the left. For example, as the exponent continues to decrease by 1, you get: $10^1 = 10.000$; $10^0 = 1.0000$. Continuing this exponential reduction and decimal shift to the left leads to the decimal fractions ($10^{-1} = .10000$; $10^{-2} = .010000$; etc.). A negative exponent of any number greater than one always represents a fractional value. This is because it is the same as 1 divided by the power of ten with the negative exponent written as a positive exponent (as shown in Table 2).

10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	Power of Ten
.10000	.01000	.00100	.00010	.00001	Decimal Value
$1/10^1$	$1/10^2$	$1/10^3$	$1/10^4$	$1/10^5$	Fraction Value

Digit Position Values for Fractional Numbers (Negative Powers of 10)

Table 2

Notice that, in Tables 1 and 2, for the whole numbers (1, 10, 100, etc.) the zeros to the right of the decimal point can be dropped without changing the value of the number. In the decimal fractions (.1, .01, .001, etc.) the zeros to the right of 1 can be dropped without changing the value of the number.

The individual symbols used to denote a quantity are called “digits,” while a group of digits arranged in a specific pattern is called a “number.” For example, 538 and 835 are two distinctly different numbers, although the same digits (3, 5, and 8) are employed. The value of each digit depends on its position in the number. For example, in the number 538, the digit 5 actually has a value of 500, the digit 3 has a value of 30, and the digit 8 has a value of 8. The sum of these values is $500 + 30 + 8 = 538$. Thus, 538 is simply a shorthand notation for $500 + 30 + 8$. Notice that the digit position or “place” corresponds to the power-of-ten magnitude, reading from left to right, in Table 1. That is, the relationship between the individual digits and the value of the overall number can be expressed in equation form as:

$$N = \dots + d_3R^3 + d_2R^2 + d_1R^1 + d_0R^0 \quad (1)$$

where N is the number, R is the radix, and d_3, d_2, d_1 , and d_0 are the digits of the number. Thus, the number 538 in the decimal system (radix, 10) may be expressed as:

$$\begin{aligned} N &= 5(10)^2 + 3(10)^1 + 8(10)^0 = 5(100) + 3(10) + 8(1) \\ &= 500 + 30 + 8. \end{aligned}$$

As another example, take the decimal number 835. Using Equation 1:

$$\begin{aligned} N &= 8(10)^2 + 3(10)^1 + 5(10)^0 = 8(100) + 3(10) + 5(1) \\ &= 800 + 30 + 5. \end{aligned}$$

A similar equation can be written for decimal fractions:

$$N = d_{-1}R^{-1} + d_{-2}R^{-2} + d_{-3}R^{-3} + \dots \quad (2)$$

Thus, the decimal number 0.372 may be expressed as:

$$\begin{aligned} N &= 3(10)^{-1} + 7(10)^{-2} + 2(10)^{-3} = 3(0.1) + 7(0.01) + 2(0.001) \\ &= 0.3 + 0.07 + 0.002 \end{aligned}$$

Combinations of Equations 1 and 2 can be used to represent numbers containing whole and fractional parts (such as 291.78).

When R is the radix of a particular number system, the R^n is a **WEIGHT**. The decimal system is indicated in Table 1 where $R = 10$. Thus, 10^n

represents the weights of the decimal system. 10^0 has a weight of one, 10^1 has a weight of 10, 10^2 possesses a weight of 100, etc. Consequently, in Tables 2 and 3, 10^{-1} has a weight of 0.1 (or $\frac{1}{10}$), 10^{-2} represents a weight of 0.01 (or $\frac{1}{100}$), etc.

Binary Numbers

Most control situations are of the “yes” or “no” variety. These situations are produced by components or circuits which have two different operating conditions, such as switches, relays, gates, and flip-flops. A two-digit (binary) system is the best notation for describing these circuit actions.

The same rules which apply to decimal numbers apply to binary numbers. But in a binary number system the radix is 2, and the only digits used are 0 and 1. The binary digits, 0 and 1, are called **BITS** (Binary digITS). Table 3 lists the decimal numbers and their binary equivalents from 0 through 50.

The whole number weights in the binary system are 1, 2, 4, 8, 16, etc. (whole number powers of the radix, 2). In Table 3, the weight 2^0 (which equals 1) has been assigned the letter A. The weight 2^1 (which is 2) was assigned the letter B, the weight 2^2 (or 4) has C, while 2^3 (or 8) has the letter D, etc. These letters, representing different weights, facilitate the implementation of tables into usable circuits, as you will see in this lesson. Of course, there are fractional binary weights also: $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, etc. (negative whole number powers of 2).

Even though only two symbols are used in the binary number system, these two symbols (0 and 1) are combined in the same manner as the ten symbols (0 through 9) used in the decimal number system.

Therefore, the binary number system is also a positional notation system. This is shown in Tables 4 and 5. Notice that, for positive exponents, as the exponent value decreases by 1, the numerical value or weight is reduced by

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	Power of Two
128	64	32	16	8	4	2	1	Decimal Value

Digit Position Weights for Positive Powers of Two

Table 4

2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	Power of Two
1/2	1/4	1/8	1/16	1/32	1/64	1/128	1/256	Fraction Value
.5	.25	.125	.0625	.03125	.015625	.0078125	.00390625	Decimal Value

Digit Position Weights for Negative Powers of Two

Table 5

a factor of 2 (divided by 2). This characteristic continues into the negative exponents as well.

Equation 1 can be used to determine the value of a binary weighted number, as well as the value of a decimal number. However, to determine the value of a binary number, the radix 10 is replaced by a radix 2. For example, the notation 101001 is:

$$\begin{aligned}
 N &= 1(2)^5 + 0(2)^4 + 1(2)^3 + 0(2)^2 + 0(2)^1 + 1(2)^0 \\
 &= 2^5 + 2^3 + 1 = 32 + 8 + 1 = 41.
 \end{aligned}$$

The binary weights in 41 are 32, 8, and 1.

As a second example, the value of 101110 is:

$$\begin{aligned}
 N &= 1(2)^5 + 0(2)^4 + 1(2)^3 + 1(2)^2 + 1(2)^1 + 0(2)^0 \\
 &= 2^5 + 2^3 + 2^2 + 2^1 = 32 + 8 + 4 + 2 = 46.
 \end{aligned}$$

The binary weights in 46 are then 32, 8, 4 and 2. Both examples check with Table 4.

Binary Number	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	Decimal Equivalent
11101	1	1	1	0	1						29
1110.1		1	1	1	0	1					14-1/2
111.01			1	1	1	0	1				7-1/4
11.101				1	1	1	0	1			3-5/8
1.1101					1	1	1	0	1		1 13/16
.11101						1	1	1	0	1	29/32

Listing of Example Binary Numbers and Decimal Equivalents (Whole and Fractional Values)

Table 6

Fractional values are shown in the binary system just as they are shown in the decimal system, by digits or bits to the right of the decimal point. Table 6

shows a variety of examples of whole numbers and fractions. Using Equations 1 and 2, the numbers listed in Table 6 have the following decimal values:

$$11101 = 1(2)^4 + 1(2)^3 + 1(2)^2 + 0(2)^1 + 1(2)^0 \\ = 16 + 8 + 4 + 1 = 29$$

$$1110.1 = 1(2)^3 + 1(2)^2 + 1(2)^1 + 0(2)^0 + 1(2)^{-1} \\ = 8 + 4 + 2 + \frac{1}{2} = 14\frac{1}{2}$$

$$111.01 = 1(2)^2 + 1(2)^1 + 1(2)^0 + 0(2)^{-1} + 1(2)^{-2} \\ = 4 + 2 + 1 + \frac{1}{4} = 7\frac{1}{4}$$

$$11.101 = 1(2)^1 + 1(2)^0 + 1(2)^{-1} + 0(2)^{-2} + 1(2)^{-3} \\ = 2 + 1 + \frac{1}{2} + 0 + \frac{1}{8} = 3\frac{5}{8}$$

$$1.1101 = 1(2)^0 + 1(2)^{-1} + 1(2)^{-2} + 0(2)^{-3} + 1(2)^{-4} \\ = 1 + \frac{1}{2} + \frac{1}{4} + 0 + \frac{1}{16} = 1\frac{13}{16}$$

$$.11101 = 1(2)^{-1} + 1(2)^{-2} + 1(2)^{-3} + 0(2)^{-4} + 1(2)^{-5} \\ = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{32} \\ = \frac{29}{32}$$

These illustrations show that the binary system, within itself, is based on the same laws which apply to the decimal system, except a radix of 2 is used. Any difficulty in applying binary notation is due to our habit of thinking in decimal numbers and the necessity of describing this binary system by using decimal numbers. In fact, binary numbers or weights are easier to work with in arithmetic; any difficulty is usually in recognizing the values of binary numbers. However, you have seen how Equations 1 and 2 can be used to convert binary numbers to decimal numbers. It is also possible to perform the reverse conversion in a simple manner.

BINARY WEIGHTS								
256	128	64	32	16	8	4	2	1

**TABULAR AID FOR CONVERTING DECIMAL
WHOLE NUMBER TO BINARY NUMBER**

Figure 2

The decimal-to-binary conversion can be performed using a table similar to the one shown in Figure 2. The first step in this conversion is to note the

highest weight in the table which is equal to or less than the number to be converted. Let us convert decimal 56. The highest weight in the table of Figure 2 which is less than 56 is 32. (The binary weights above 32 are all greater than 56, and the weights below 32 are not the highest numbers which are less than 56.) Now place a 1 in the block under 32 and subtract 32 from 56 to give 24. This is shown in Figure 3A. Now note the highest weight in the table which is equal to or less than the answer of the first subtraction (24). The highest weight in the table which is less than 24 is 16. Therefore, place a 1 in the block under 16 and subtract 16 from 24 to give 8. This is shown in Figure 3B. Now note the highest weight in the table that is equal to or less than 8. This weight in the table is 8. Therefore, place a 1 under the 8 and subtract 8 from 8 to leave 0. This is shown in Figure 3C.

56	256	128	64	32	16	8	4	2	1
-32									
24				1					

A

24	256	128	64	32	16	8	4	2	1
-16									
8				1	1				

B

8	256	128	64	32	16	8	4	2	1
-8									
0				1	1	1			

C

	256	128	64	32	16	8	4	2	1
				1	1	1	0	0	0

D

STEPS IN THE CONVERSION OF DECIMAL 56 TO THE EQUIVALENT BINARY NUMBER 1111000

Figure 3

In Figure 3C, there are 1's in the 32, 16 and 8 columns. No more subtractions can be performed since the last answer was 0. However, the binary number is not yet complete. To complete the binary number there must be symbols in the 4, 2, and 1 columns so the 1's will be in the proper positions.

Since the decimal values (or weights) 4, 2 and 1 are not needed to make up 56, place 0's in the corresponding columns. This is shown in Figure 3D. Thus, the binary equivalent of the decimal number 56 is 111000. To check this answer you can reconvert this binary number to its decimal equivalent, using Equation 1.

Decimal numbers with decimal points may be converted to binary numbers by using a table similar to Table 6. In converting a decimal number with a decimal point to its binary equivalent, first convert the digits to the left of the decimal point; then convert the digits to the right of the decimal point. For example, suppose you wish to convert the decimal number 3.625 to its binary equivalent. First, convert the digit to the left of the decimal point, which in the number 3.625 is 3. Since the weight 2 from 3 leaves 1, and the weight 1 from 1 leaves 0, place 1's in both the 2 and 1 columns of the table as shown in Figure 4A.

8	4	2	1		$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$
		1	1	.				

A

.625	8	4	2	1		$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$
-.5			1	1	.	1			
.125									

B

.125	8	4	2	1		$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$
-.125			1	1	.	1		1	
0									

C

	8	4	2	1		$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$
			1	1	.	1	0	1	

D

STEPS IN THE CONVERSION OF DECIMAL 3.625
TO THE EQUIVALENT BINARY NUMBER 11.101

Figure 4

Now convert the digits .625 to their binary equivalent using the same procedure. First, note the highest number (or weight) in the table which is equal to or less than .625. In the table of Figure 4 this weight is $\frac{1}{2}$ or .5. Now place a 1 in the block under $\frac{1}{2}$ and subtract .5 from .625, which leaves .125 (as shown in Figure 4B). Now, note the highest weight in the table which is equal to or less than .125. This weight is $\frac{1}{8}$, or .125. Place a 1 in the block under $\frac{1}{8}$ and subtract .125 from .125 which leaves 0 (as shown in Figure 4C). No more subtractions can be performed because the last answer was 0. To complete the binary number, place a 0 in the block under $\frac{1}{4}$ as shown in Figure 4D. Thus, 3.625 in the decimal system equals 11.101 in the binary system.

The binary number system is a code in itself. Any decimal number can be represented by a group of binary digits (1's and 0's) or bits. The **NATURAL BINARY CODE** is a very efficient code. Compared to most other codes, it uses fewer bits to represent a given decimal number. However, the natural binary code has one major disadvantage. Conversion of decimal numbers to their natural binary codes cannot be accomplished with simple electronic or mechanical conversion devices. Complex electronic or mechanical conversion devices involving timing signals, enabling or disabling gates, and counters are required.

The ease with which arithmetic operations can be performed with the natural binary code often overshadows the conversion problem. Simple arithmetic units may be used in computing systems utilizing the natural binary code as their machine language. Most often the computing systems using the natural binary code are those designed for scientific applications where many calculations must be performed. Digital systems, in general, seldom use the natural binary code. Instead, they use modified forms of this code.

Binary-Coded Decimal Numbers

A direct modification of the natural binary code is the **BINARY-CODED DECIMAL (BCD) CODE**, also known as the 8-4-2-1 code. In this code, each digit of a decimal number is represented by a group of four binary digits or bits called a code group. Depending upon the digit being coded, the four-bit code group will be one of the natural binary codes for the decimal numbers 0 to 9. Notice in Table 3 that the natural binary codes for the decimal numbers 0 to 7 require less than four bits. If zeros are added to the left of these code groups, they can be made to contain four bits and yet retain their original values. Using this procedure, the natural binary codes for the decimal numbers 0 to 9 are as shown in Table 7. These four-bit code groups may now be used to construct the BCD code for any decimal number. Note, for example, that A has a weight of 1 while D has a weight of 8.

DECIMAL	BCD CODE			
	D	C	B	A
	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Basic BCD Code

Table 7

Let's find the BCD code for the decimal number 56. With zeros added to form four-bit code groups, the natural binary codes for 5 and 6 are 0101 and 0110, respectively. Combining these two code groups yields 0101 0110, which is the BCD for 56. The first group of four bits (0101) represents 5, and the second group (0110) represents 6. A space has been left in between the two code groups so that the numbers they represent can be readily determined. In a digital system, however, this space would not be used and the BCD code for 56 would appear as 01010110. Actually, in a particular digital system, zeros would be added to the left of 01010110 to fill the system to capacity.

As another example, let's find the BCD code for the decimal number 278. The natural binary codes for 2, 7 and 8 are 0010, 0111, and 1000, respec-

tively, with zeros added to form four-bit code groups. Combining these code groups yields 0010 0111 1000, which is the BCD code for 278. For a third example, consider the decimal number 892. The natural binary codes for 8, 9 and 2 are 1000, 1001, and 0010, respectively. Combining these code groups yields 1000 1001 0010, which is the BCD code for 892.

Conversion from a binary-coded decimal number back to a decimal number is quite simple. All that is necessary is that each four-bit code group be converted back to the decimal digit it represents. This may be readily accomplished with the aid of Table 7. To illustrate the process, consider the BCD number 0101 0011 1001. Using Table 7, the code group 0101 represents 5; 0011 represents 3; and 1001 represents 9. Combining the digits in the same order as their respective code groups in the coded number yields the decimal number 539. As another example, let's convert the BCD number 0100 1000. Using Table 7, the code 0100 represents 4, and 1000 represents 8. Combining the digits yields 48.

The ten four-bit code groups in Table 7 do not represent all possible ways that 0 and 1 can be arranged in a four-bit combination (there are 2^4 or 16). The groups 1010, 1011, 1100, 1101, 1110 and 1111 could be formed, but are not shown in Table 7. As far as the BCD code is concerned, these groupings are meaningless. They are called **FORBIDDEN COMBINATIONS**. If one of them should appear in a digital system using the BCD code, it would be an error.

You would do well to study carefully the digital equivalents of the numbers 0 through 9 (0000 through 1001) because, in most cases, you will be making good use of that information.

The Excess-3 Code

Another binary-coded decimal code is the **EXCESS-3 CODE**, commonly indicated as the XS-3 code. The XS-3 code is similar to the BCD code in the respect that four bits are used to represent each digit in the decimal number being coded. The XS-3 code of a digit is the BCD code of the digit plus 3. This is the reason why this code is called the excess-3 code. The XS-3 is NOT a weighted code, and Equations 1 and 2 do not apply because decimal zero does not equal zero in XS-3.

The XS-3 code is formed from the BCD code by adding the BCD code for 3 to the BCD code for the decimal digit being coded. The addition is performed using the rules of binary addition. For the moment, you need to be familiar only with the resulting code groups, listed in Table 8.

DECIMAL	XS-3
0	0011
1	0100
2	0101
3	0110
4	0111
5	1000
6	1001
7	1010
8	1011
9	1100

The Basic Excess-3 Code

Table
8

As an example of converting a decimal number to the XS-3 code, let's convert the decimal number 27. The XS-3 codes for 2 and 7 are 0101 and 1010, respectively, from Table 8. Combining these code groups in the digit-order associated with the decimal number, we have 0101 1010. You can see that the procedure is similar to that for BCD coding—we just use the XS-3 code groups from Table 8. However, we can make use of Table 7 to change the decimal equivalent numbers to XS-3 code numbers by adding 3 to each decimal number and then use Table 7. For the 2 in 27, add 3: $2 + 3 = 5$, whose natural binary code is 0101, and for 7, adding 3 produces 10, whose code is 1010. Again we have 0101 1010.

Conversion from an XS-3 coded decimal number back to the decimal number is quite simple. All that is necessary is that each four-bit code group be converted back to the decimal digit it represents. This may be readily accomplished with the aid of Table 8. Take, for example, the XS-3 coded number 0110 1010 0100. Using Table 8, 0110 represents 3; 1010 represents 7; and 0100 represents 1. Combining the digits yields 371. As another example, consider the XS-3 coded number 0101 1000. From Table 8, 0101 represents 2 and 1000 represents 5. Combining the digits yields 25.

The Gray Code

A code which is often used in conjunction with digital systems is the **GRAY CODE**. The Gray code received its name from the man who developed it, Dr. Gray of Bell Laboratories. The Gray code is commonly used in analog-to-digital converters, such as devices which convert the angular position of a shaft into a binary-coded set of electric signals.

The Gray code can be formed from the natural binary code. To find the Gray code for a given number, follow the four steps listed below:

1. Find the natural binary code for the number to be coded.
2. Write the number from Step 1 down twice, one over the other, as is done in arranging numbers which are to be added.
3. Shift the bottom number one place to the right and disregard the rightmost digit in the number.
4. Add the two numbers together by the rules shown in Table 9. The sum produced by the process is the Gray code for the decimal number being coded.

0	+	0	=	0
1	+	0	=	1
0	+	1	=	1
1	+	1	=	0

Addition Rules
for Gray Code
Development

Table
9

For example, let's find the Gray code for the decimal number 25. From Table 3 the natural binary code for 25 is 11001. The process of adding the number to itself to obtain the Gray code is shown below:

Step 1 11001 The Natural Binary Code for 25

Step 2 11001
+ 11001

Step 3 11001
+ 1100~~1~~ Shift and Disregard Right-most Digit

Step 4 Use the rules from Table 9:

11001
+ 1100
10101 The Gray Code for 25.

1101
140
1011

As another example, find the Gray code for the decimal number 13. From Table 3 the natural binary code for 13 is 1101. The addition yielding the Gray Code is shown below:

1101 Natural Binary Code for 13
+ 110~~1~~ Shift and Disregard Right-most Digit
1011 Gray Code for 13.

Following the previously described procedure, the Gray code for any decimal number can be readily found. Table 10 shows the Gray Code for several decimal numbers.

DECIMAL	GRAY CODE	DECIMAL	GRAY CODE
0	0000	13	1011
1	0001	14	1001
2	0011	15	1000
3	0010	16	11000
4	0110	17	11001
5	0111	18	11011
6	0101	19	11010
7	0100	20	11110
8	1100	21	11111
9	1101	22	11101
10	1111	23	11100
11	1110	24	10100
12	1010	25	10101

Partial Listing of the Gray Code

Table 10

Gray coding has the characteristic of changing only one bit in going from one number to the next succeeding number. For example, 3 is 10 and 4 is 110 in Gray code. Only one bit changes in going from 3 to 4. However, in the natural binary code 3 is 11 and 4 is 100. Three bits change in progressing from 3 to 4 in the natural binary code. Because the Gray code only changes one bit in progressing to the next succeeding number, it is often called the **CYCLIC CODE**; it is not a weighted code, and Equations 1 and 2 do not apply to the Gray code.

NATURAL BINARY CODE	BCD CODE	XS-3 CODE	GRAY CODE	BINARY DIGITS			
				D	C	B	A
				8	4	2	1
0	0	-	0	0	0	0	0
1	1	-	1	0	0	0	1
2	2	-	3	0	0	1	0
3	3	0	2	0	0	1	1
4	4	1	7	0	1	0	0
5	5	2	6	0	1	0	1
6	6	3	4	0	1	1	0
7	7	4	5	0	1	1	1
8	8	5	15	1	0	0	0
9	9	6	14	1	0	0	1
10	-	7	12	1	0	1	0
11	-	8	13	1	0	1	1
12	-	9	8	1	1	0	0
13	-	-	9	1	1	0	1
14	-	-	11	1	1	1	0
15	-	-	10	1	1	1	1

Comparison of Codes to Binary Digits

Table 11

Table 11 compares the Natural Binary Code, BCD code, XS-3 code, Gray code, and binary digits with weights 1, 2, 4 and 8. However, the weights of the binary digits apply only to the Natural Binary and BCD codes.

The following Practice Exercise questions cover the subjects which you have just studied. They are:

NUMBER SYSTEMS

DECIMAL NUMBERS

BINARY NUMBERS

BINARY-CODED DECIMAL NUMBERS

THE EXCESS-3 CODE

THE GRAY CODE

1. A number system composed of only the digits 0, 1, 2 and 3 has a radix of _____.
2. Write 1,000,000 as a power of ten.
3. Use Equation 1 to write 7,842 as a sum of powers of ten.
4. Use Equation 2 to write 0.65 as a sum of powers of ten.
5. Use Equations 1 and 2 to write 691.28 as a sum of powers of ten.
6. Convert the following binary numbers to decimal numbers:
(a) 101001001 (b) 10101010 (c) 110001 (d) 110101 (e) 1000000 (f) 1011.001
7. Convert the following decimal numbers to binary numbers:
(a) 53 (b) 64 (c) 71 (d) 102 (e) 5.5 (f) 12.375
8. What is the relationship between weights and radices?
9. What binary weights are in the binary number for 57?
10. In the BCD code, each digit of a decimal number is represented by a three-bit code group. True or False?
11. What is the BCD code for the decimal number 1975?
12. The code 1000 0110 1001 represents what decimal number?
13. The code group 1101 represents the decimal number 13 in the BCD code. True or False?
14. The XS-3 code is formed by adding the BCD code of the digit being coded to the BCD code of 3. True or False?

15. What is the XS-3 code for the decimal number 173?

16. The XS-3 code groups 1010 1100 0110 represent what decimal number?

DECODING LOGIC

The information processed by a digital system will be in the form of one of the binary codes. However, when the results of the process are read-out of the system, we wish to have them in the form of decimal numbers for easy interpretation. The logic circuits used to perform this conversion are known as decoding logic. The circuits themselves may use diode logic, diode-transistor logic, resistor-transistor logic or any of the many logic circuit types which you are already familiar with. The decoding arrangement always will be cascade or parallel combinations of the basic logic functions (AND, OR, NOT) or of various combinations of the NAND, NOR EXCLUSIVE-OR, etc., logic functions. In general, decoding logic takes a number of input code bits (such as the four binary digits in a BCD code group) and converts them into one of several possible outputs (such as one of the ten decimal-digit values which can be represented by a single BCD code group). Such an example is shown in Figure 5. A BCD code group representing decimal 6 has resulted from a particular digital process. It is

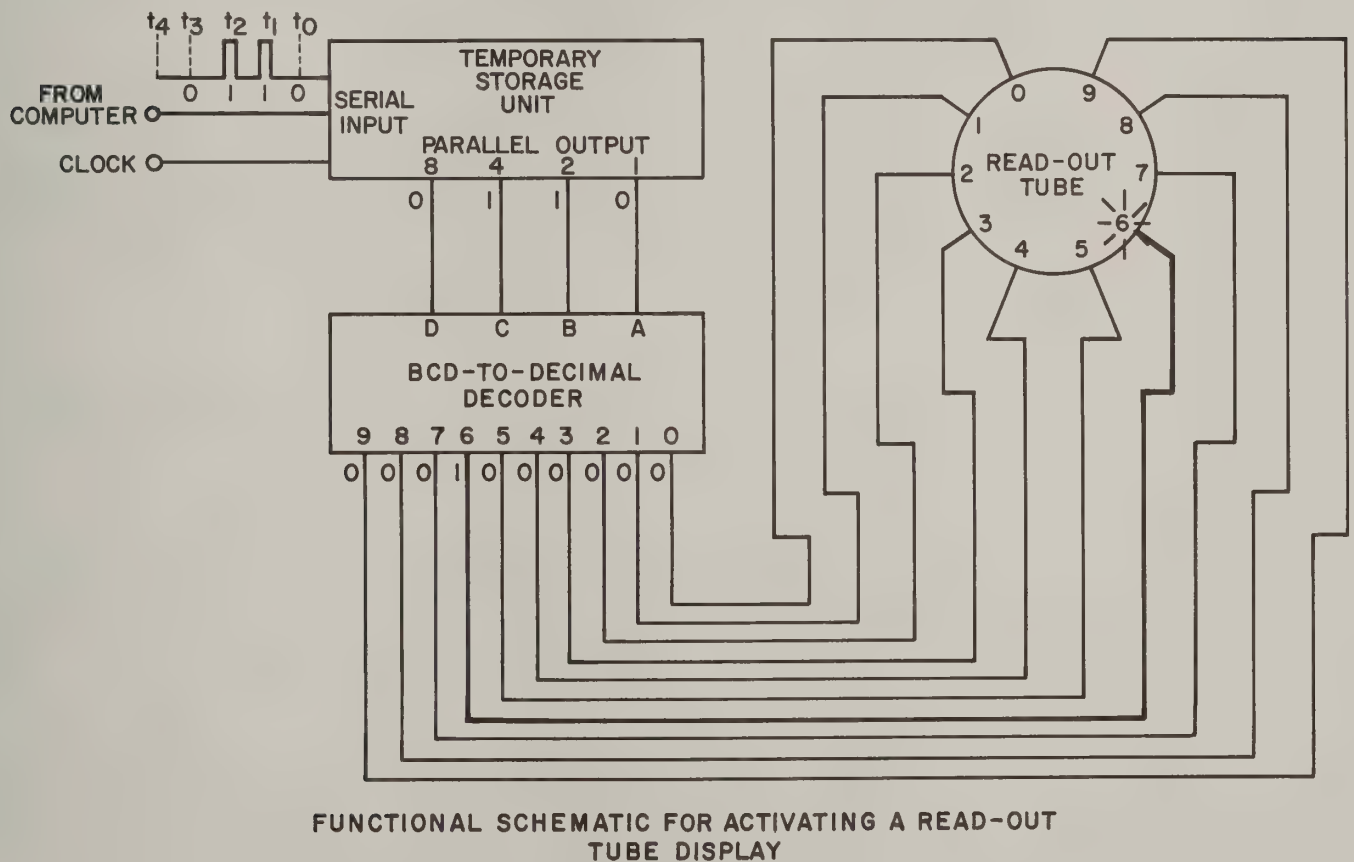


Figure 5

temporarily stored and the four stored binary digits are used to feed a BCD-to-decimal decoder. The decoder activates one of ten output signals, in this case the output signal required to light up "6" on the Nixie read-out tube.

The pulses entering the temporary storage unit (also known as a serial-to-parallel converter) are in dynamic logic form since they enter one at a time (the t_0 pulse enters first and has the lowest weight of 2^0 , or 1). The pulses enter in what is often called serial form and appear at the output in parallel form (static logic). Only four pulses can be received at a time because there are only four bistables in the temporary storage unit, as indicated by the four output terminals, which have weights of 1, 2, 4 and 8 from right to left. These output terminals are assigned letters A, B, C and D (in the decoder) respectively, as shown in Figure 5.

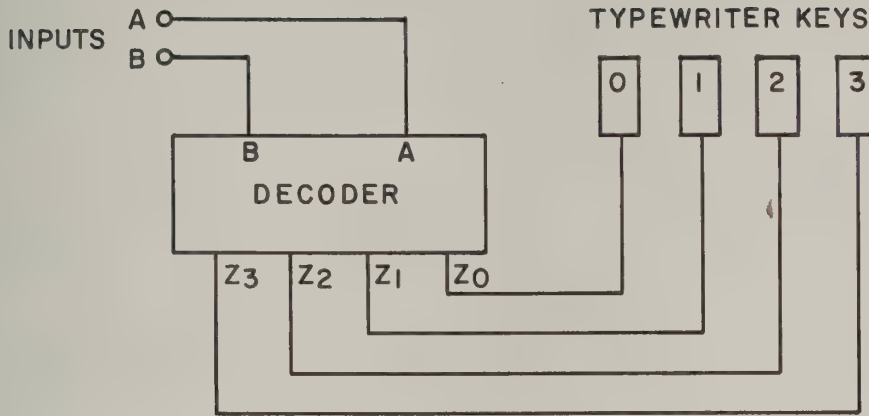
The t_0 pulse enters the temporary storage unit first and the t_3 pulse is the last of the four pulses to enter. Usually, there is a clock pulse at the same frequency of the input pulses that prepares the bistables in the temporary storage unit for the entry of each input pulse. After all four pulses have entered the temporary storage unit, the logic 0 at the weight 1 output results from the t_0 pulse; the weight 2 output has a logic 1 because of the t_1 pulse; the weight 4 output logic 1 results from the pulse at t_2 ; and then the logic 0 at the weight 8 output was produced by the t_3 pulse. Sometimes there is a pulse at t_4 called the read-out pulse which transfers the parallel output of the temporary storage unit to the BCD-to-decimal decoder so that the decoder can illuminate the read-out tube or device.

The BCD-to-Decimal Decoder has the inputs A, B, C, and D where A is the least significant input. The outputs are 0 through 9. Some manufacturers identify the outputs as Q0 through Q9.

Some BCD-to-decimal decoders (especially in the IC form) have a logic 0 at the desired output and logic 1's at all of the rest. This situation results from the output circuits being NAND and NOR gates instead of AND and OR gates. Thus, some manufacturers identify the outputs as $\bar{Q}0$ through $\bar{Q}9$.

Many decoders of this type may be used in a single digital system. For example, there are several desk-top electronic calculators on the market which can display 16-digit results in side-by-side Nixie tube or LED (light-emitting diode) arrangements. Such devices utilize sixteen of the circuit arrangements shown in Figure 5. The circuits are connected in parallel—one decoder arrangement for each of the sixteen decimal digits.

AND Expression Decoding



EXAMPLE DECODER FOR ACTIVATING
ONE OF FOUR TYPEWRITER KEYS
WITH TWO-BIT INPUT CODE

Figure 6

We will use a very simple example to illustrate one of the basic processes for performing the decoding function. The resulting logic arrangement is known as an **AND EXPRESSION DECODER**.

Decimal Number	Inputs		Outputs			
	B	A	Z ₃	Z ₂	Z ₁	Z ₀
	$2^1 = 2$	$2^0 = 1$				
0	0	0	0	0	0	1
1	0	1	0	0	1	0
2	1	0	0	1	0	0
3	1	1	1	0	0	0

Decoding Truth Data for System shown in Figure 6

Table 12

Assume we wish to drive one of four typewriter keys representing the decimal digits 0, 1, 2 and 3 shown in Figure 6 (a Two-Line Binary to 1-of-4 decoder). The decoder must provide one of four outputs: Z_0 , Z_1 , Z_2 or Z_3 . The decimal numbers 0, 1, 2 and 3 can be represented with a two-bit natural binary code, as listed in Table 12. Note: In designing truth tables you can label inputs and outputs reading from right to left or left to right. In this lesson both labeling systems are used.

A	B	C	Z_0	Z_1	Z_2
C	B	A	Z_2	Z_1	Z_0

Let's use the truth data of Table 12 to write the AND expression for each output:

$$\begin{aligned} Z_0 &= \bar{A} \cdot \bar{B} \\ Z_1 &= A \cdot \bar{B} \\ Z_2 &= \bar{A} \cdot B \\ Z_3 &= A \cdot B \end{aligned}$$

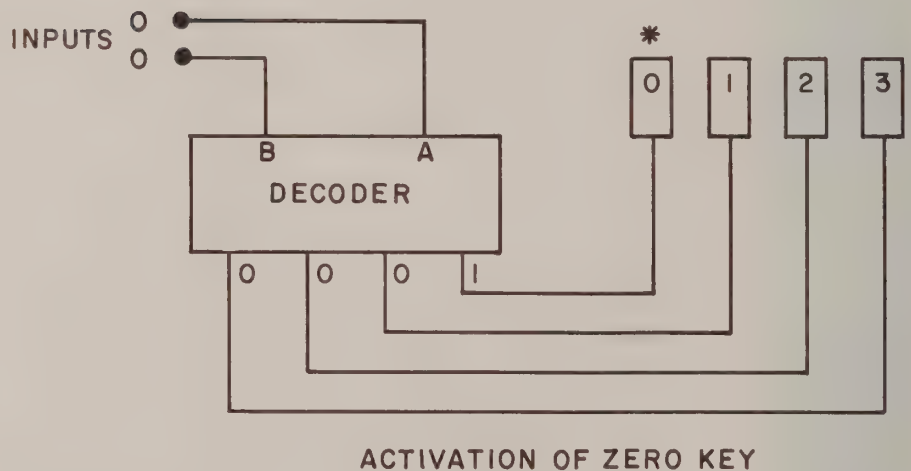


Figure 7

The decoder must take two inputs (the value of each of the binary digits in the two-bit natural binary code) and must produce one binary one output and three binary zero outputs in order to drive one of the typewriter keys. An example is shown in Figure 7 where the input is 00 and only the Z_0 output is 1, causing only the zero-key to be activated.

Figure 8 illustrates the logic networks which can be used to implement each of the four AND expressions listed in the preceding paragraph where

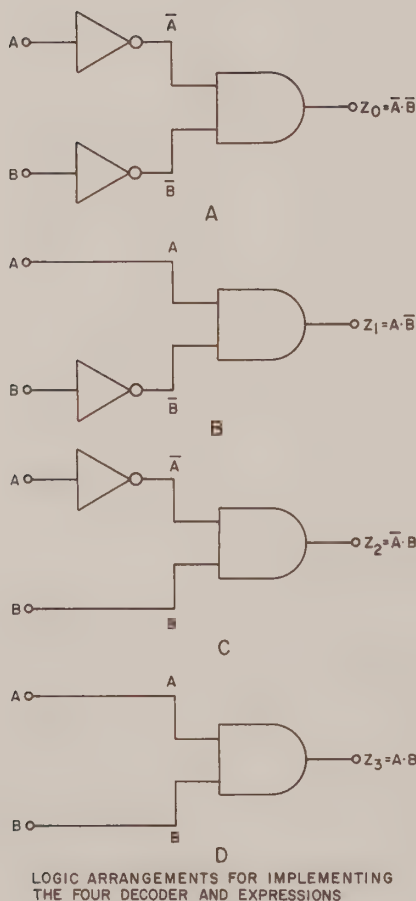
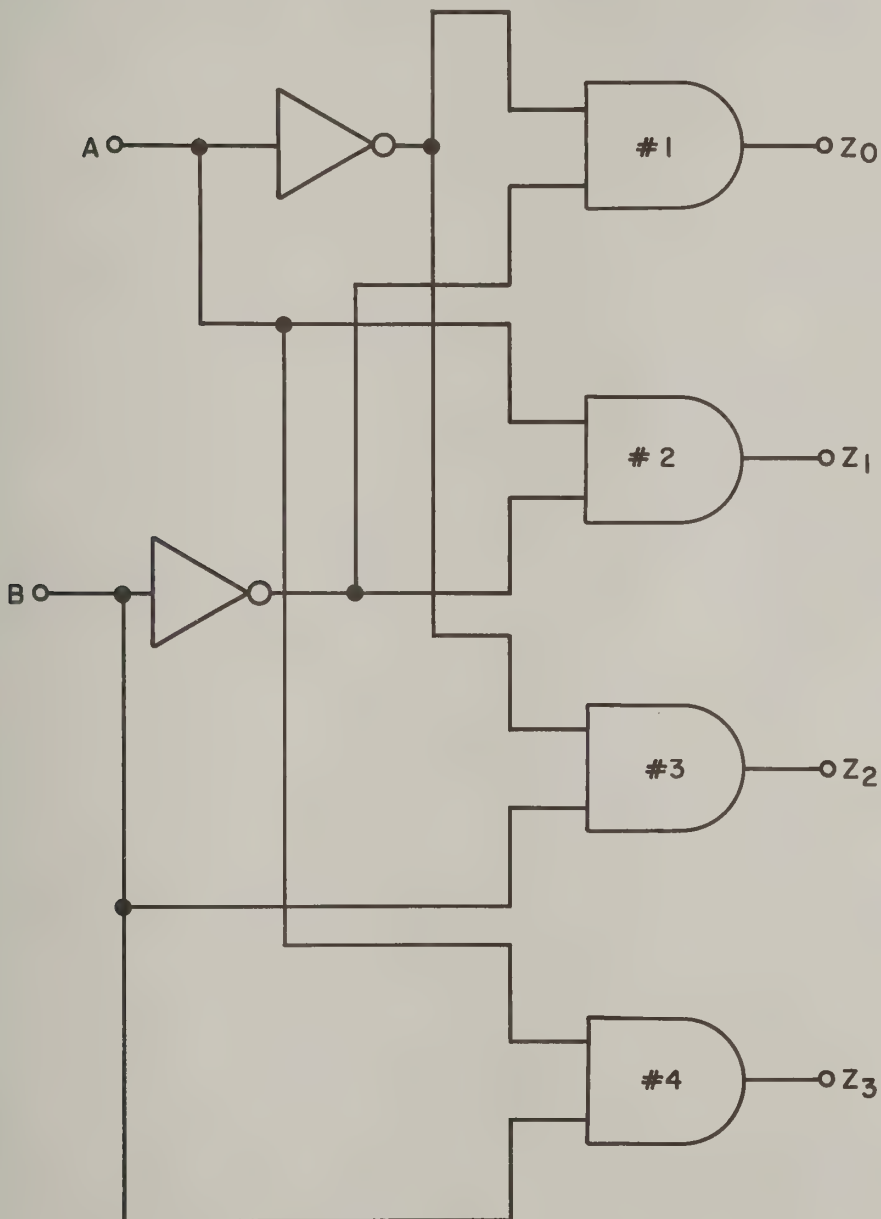


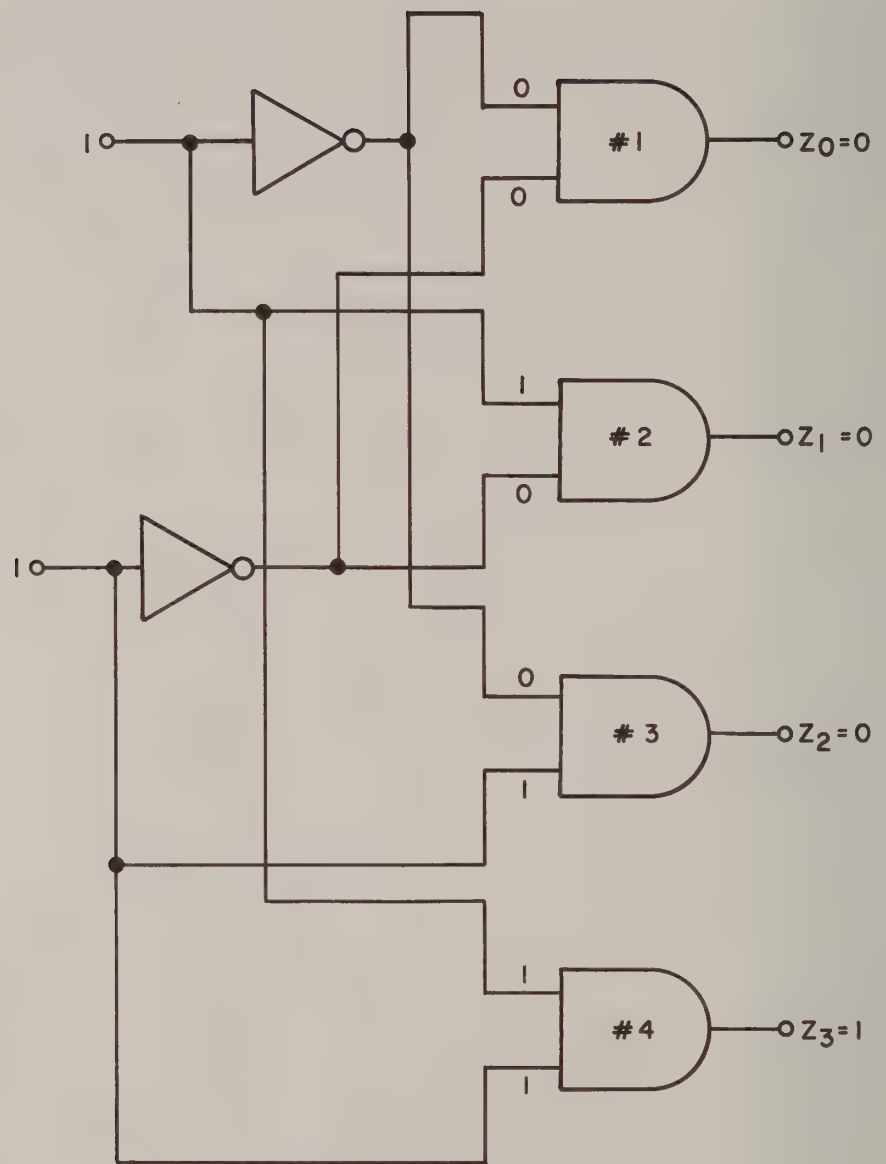
Figure 8

A has a weight of 1 and B has a weight of 2. They are simple combinations of AND and NOT functions. However, for a practical decoder, we would combine the two inputs in such a way as to drive all the logic elements in a single arrangement. The result is shown in Figure 9.



FINAL DECODER LOGIC ARRANGEMENT
OF THE CIRCUITS IN FIGURE 8

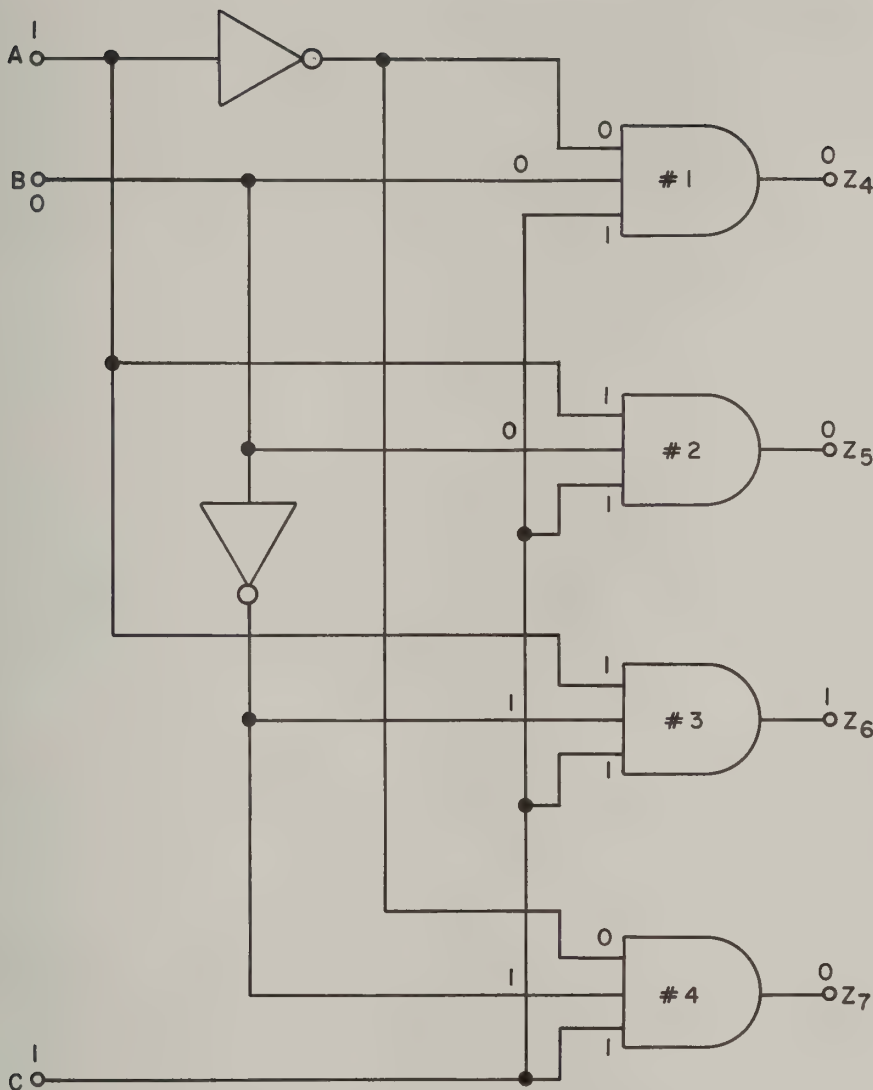
Figure 9



DECODER OPERATION WITH
BINARY 11 AS INPUT
(FOR DECIMAL 3)

Figure 10

You can satisfy yourself that the decoder logic in Figure 9 performs the desired function, by inputting the different codes from Table 12 and developing the output result on a level-by-level basis. For example, in Figure 10, the binary code 11 (for decimal 3) is inputted. The logic values at various points in the circuit are also shown. The result is a 1-output for Z_3 and a 0-output for Z_0 , Z_1 , and Z_2 ; therefore, only the decimal-3 key is activated.



AND EXPRESSION DECODER FOR
THREE-BIT GRAY CODE (FROM TABLE 13)

Figure 11

Figure 11 illustrates an AND expression decoder for converting a three-bit Gray code for decimals 4, 5, 6, and 7 into a single output representing one of these decimal digits. The truth data is shown in Table 13. The operation of the decoder is verified for the Gray code group 101 (decimal 6).

We cannot show all possible AND expression decoders. There are as many possibilities as there are codes and code groups. We have only attempted to acquaint you with the development of this type decoder in order that you might gain a better understanding of the decoding process. In general, if

Decimal Number	Inputs			Outputs			
	C	B	A	Z ₇	Z ₆	Z ₅	Z ₄
4	1	1	0	0	0	0	1
5	1	1	1	0	0	1	0
6	1	0	1	0	1	0	0
7	1	0	0	1	0	0	0

Decoding Truth Table for a Three-Bit Gray Code for
Decimal 4, 5, 6 and 7

Table 13

a decoder uses AND gates with as many inputs as there are code bits, the arrangement requires one AND gate for each output. A BCD-to-decimal maxterm decoder could be constructed with ten four-input AND gates. However, when more than three inputs are used (more than three code-group bits), there is usually enough commonality between two or more of the AND expressions to allow simplification of these expressions. This, in turn, can result in simpler logic arrangements.

Pyramid Decoding

Decimal Number	Inputs			Outputs							
	C $2^2 = 4$	B $2^1 = 2$	A $2^0 = 1$	Z ₇	Z ₆	Z ₅	Z ₄	Z ₃	Z ₂	Z ₁	Z ₀
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
2	0	1	0	0	0	0	0	0	1	0	0
3	0	1	1	0	0	0	0	1	0	0	0
4	1	0	0	0	0	0	1	0	0	0	0
5	1	0	1	0	0	1	0	0	0	0	0
6	1	1	0	0	1	0	0	0	0	0	0
7	1	1	1	1	0	0	0	0	0	0	0

Truth Data for the Conversion of a Three-Bit Natural Binary Code for
Decimal 0 through 7

Table 14

Table 14 presents the three-bit natural binary codes for the decimal numbers 0 through 7. These eight code groups can be categorized into four sets with common A and B terms. For example, Z_0 and Z_4 have common A and B terms: $\bar{A} \cdot \bar{B} \cdot \bar{C}$ and $\bar{A} \cdot \bar{B} \cdot C$. These two codes can be decoded by the arrangement shown in Figure 12. Similar arrangements for the pairs: Z_1 and Z_5 , Z_2 and Z_6 , and Z_3 and Z_7 are shown in Figure 13. The four sets are combined into a single decoder arrangement in Figure 14. This figure also verifies the decoder operation for the input 011.

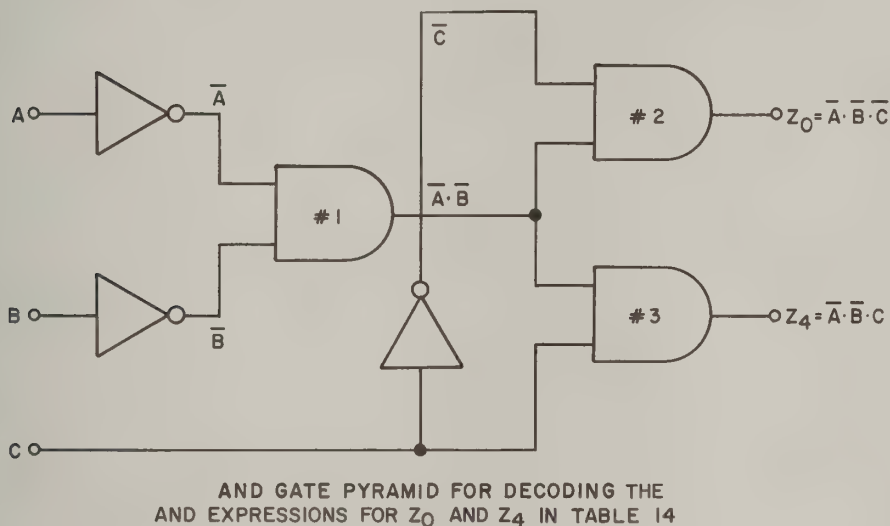
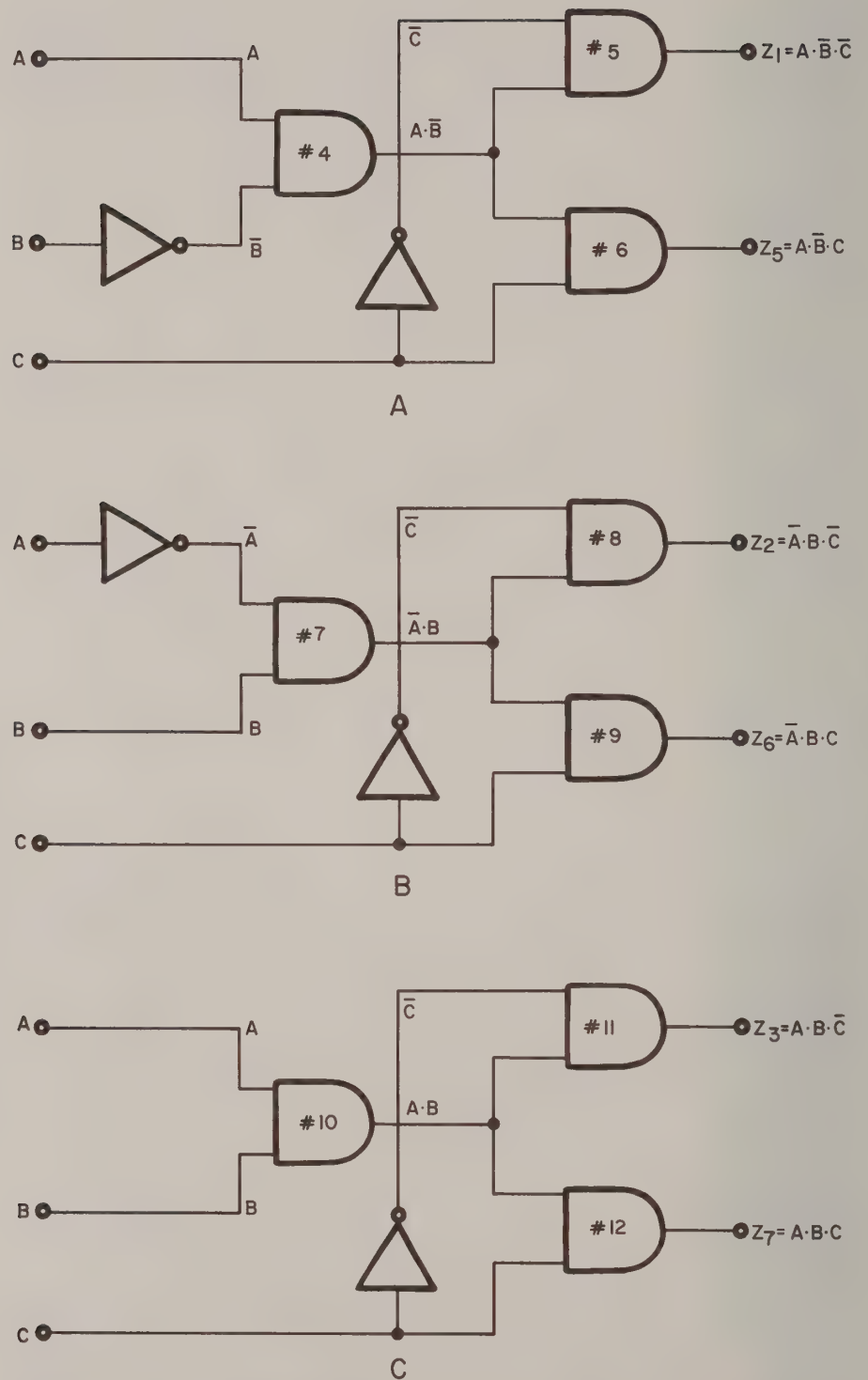


Figure 12

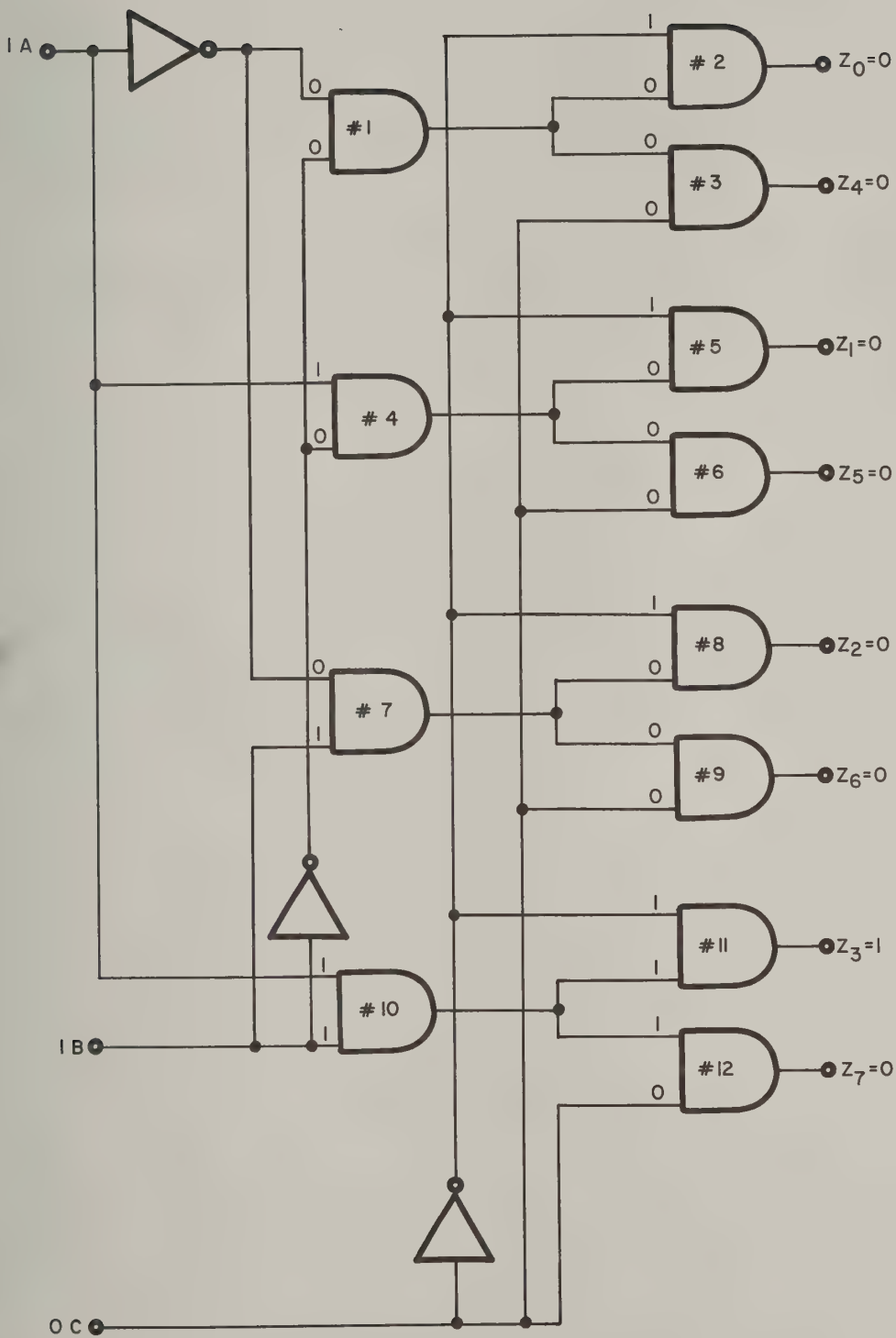
The basic arrangement in Figures 12 through 14 is known as a **PYRAMID DECODER**. An AND expression decoder could be developed to perform the same conversion. With three-input AND gates, such a decoder would require eight AND gates. This is four less than shown in Figure 14. However, the gates in Figure 14 are all two-input gates, resulting in a much simpler interconnection. The pyramid decoder also offers another significant advantage. It can be expanded to accept larger code groups simply by pyramiding another level of AND gates, using the outputs already available.

The three-position decoder in Figure 14 is expanded to a four-bit decoder in Figure 15. Each output from the three-bit decoder is pyramided into two new AND gates, along with the new input state and its inverse (D and $\bar{D}$). The sixteen gates in Figure 15 are the decimal numbers 0 through 15. The inputs are the four-bit natural binary code groups for these numbers (see Table 3). This is verified in Figure 15 for the code group 1101 (decimal 13). Note that, in this particular decoder circuit, the input (1101) is read from bottom to top (inputs D , C , B and A , respectively). By rearranging the circuitry, it could be read from top to bottom.



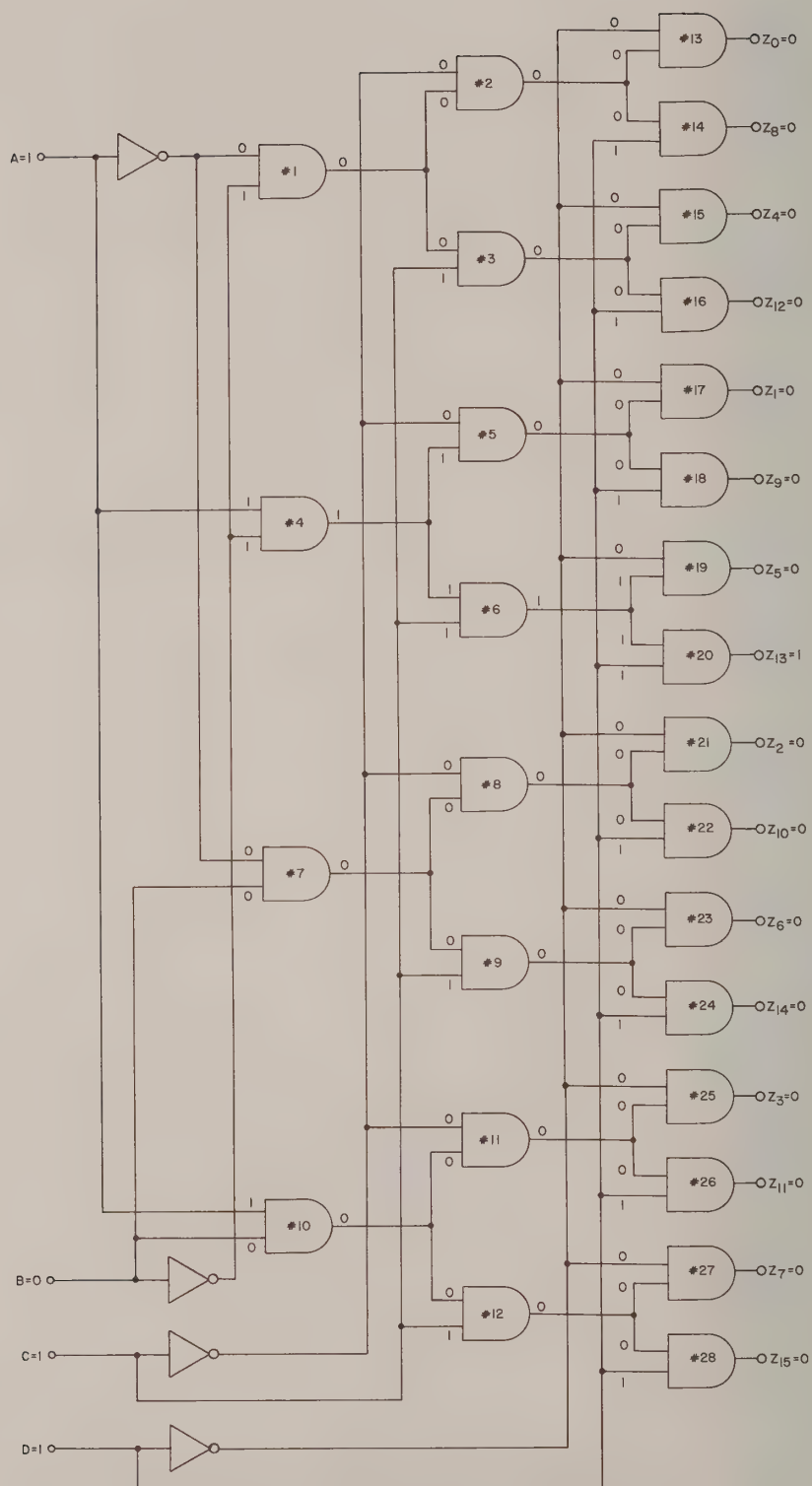
AND GATE PYRAMIDS FOR DECODING THE Z_1 AND Z_5 , Z_2 AND Z_6 , AND Z_3 AND Z_7 AND EXPRESSIONS IN TABLE 14

Figure 13



COMPOSITE PYRAMID(3 INPUT BINARY TO 1 OF 8) DECODER FOR
AND EXPRESSIONS IN TABLE 14

Figure 14



PYRAMID DECODER FOR FOUR-BIT BINARY CODE (FOUR LINE BINARY TO ONE OF SIXTEEN)

Figure 15

Since the BCD code groups are the same as the natural binary code groups for decimal numbers from 0 through 9, the pyramid decoder in Figure 15 also can be used as a BCD-to-decimal decoder. However, since the code groups 1010, 1011, 1100, 1101, 1110 and 1111 (decimals 10 through 15) never occur with a BCD code input, the AND gates for these outputs (Z_{10} through Z_{15}) can be deleted.

In IC form, it is very likely that the output gates will be NAND's (or NOR's) so that the desired output will be at a logic 0 and all of the rest will be at logic 1. If it is needed, each output can go to an inverter so that one output is a logic 1 and the rest a logic 0. Many, perhaps most, IC decoders are in the transistor-transistor logic form.

Matrix Decoding

The logic arrangement shown in Figure 16 is known as a **MATRIX DECODER**. It is a more "balanced" arrangement than the pyramid decoder and usually results in the use of fewer AND gates. For example, the decoder in Figure 16 performs the same function as the decoder in Figure 15. It will convert the four-bit natural binary code groups into the decimal numbers 0 through 15. Yet, the matrix has four less AND gates than the pyramid. The inverters in Figure 16 are not shown; four would be needed (one for each input).

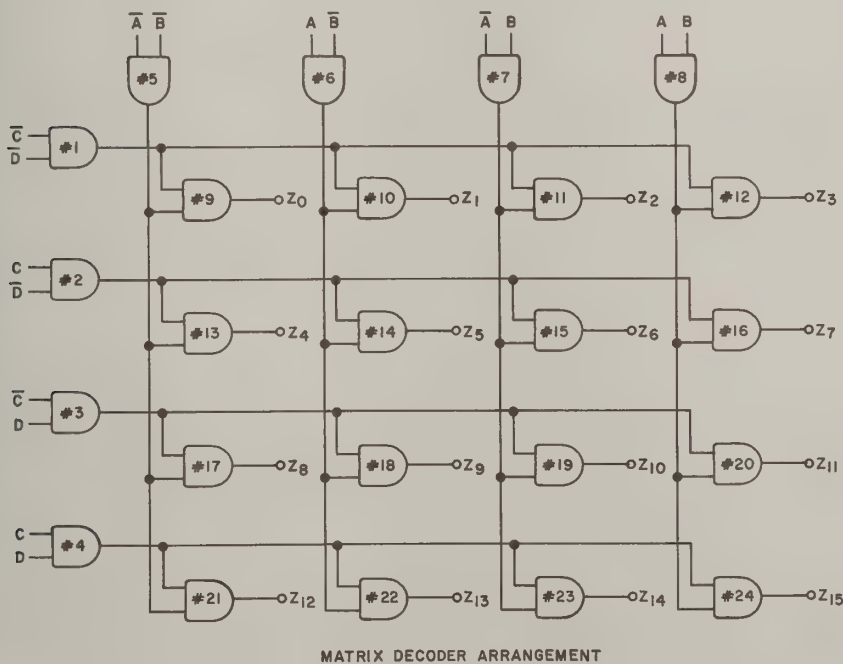


Figure 16

The pyramid decoder is generally used with transistor logic circuits. The matrix decoder may also be implemented with transistor circuits; however, it is most often found as the arrangement used with diode decoders.

The matrix decoder of Figure 16 can also be used as a BCD-to-decimal decoder. As with the equivalent pyramid decoder, the AND gates for Z_{10} through Z_{15} should be deleted.

Special Decoder Logic

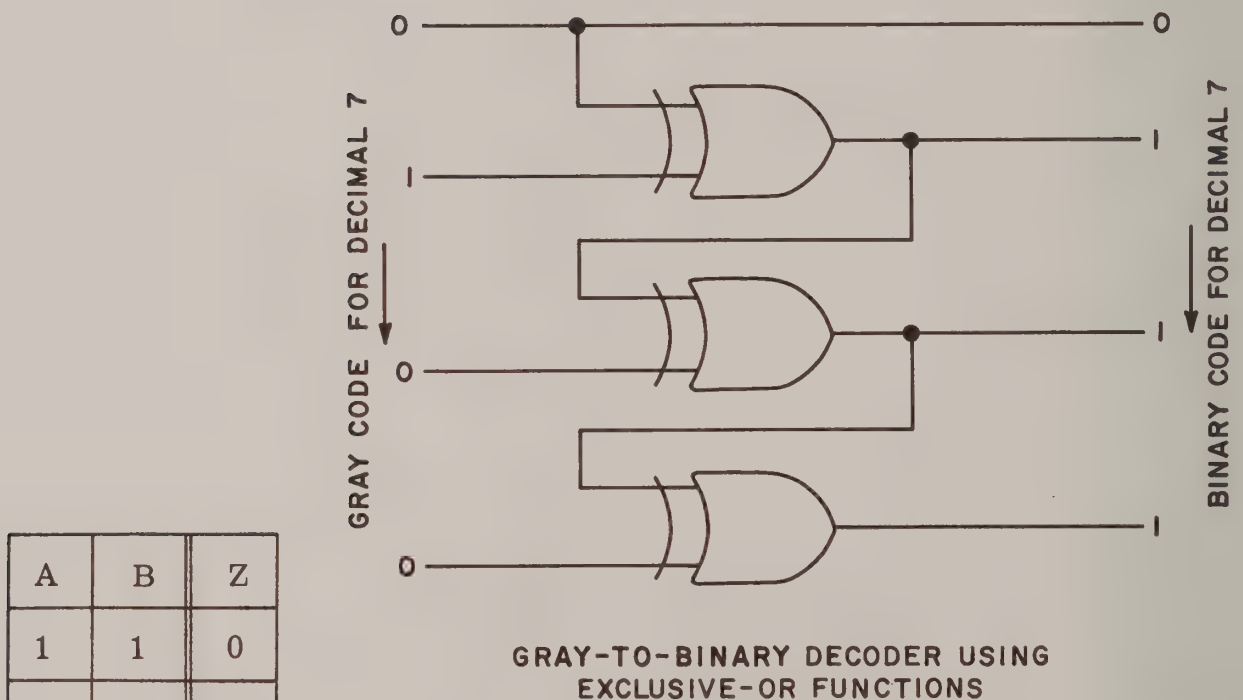
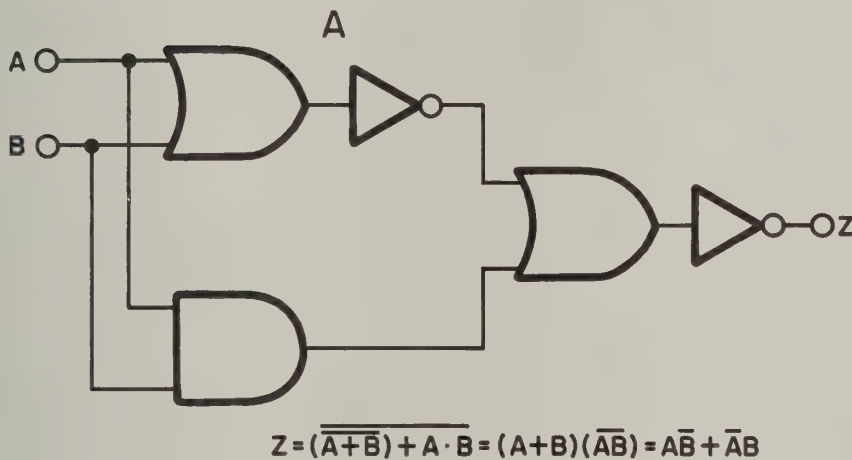
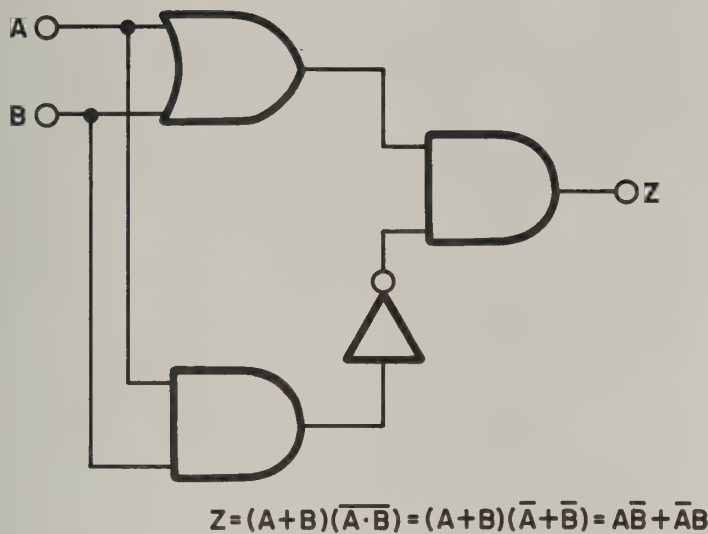


Figure 17

Truth Data for EXCLUSIVE-OR

Table 15

Many decoder arrangements can be formed by using more complex logic functions. As an example, Figure 17 illustrates an arrangement of EXCLUSIVE-OR functions used to convert Gray code groups into natural binary code groups. This arrangement uses feedback from two of the EXCLUSIVE-OR outputs. The EXCLUSIVE-OR truth data are listed in Table 15. You may use these data to verify the conversion of decimal 7 (Gray Code 0100 to natural binary code 0111) shown in Figure 17. Note that here the inputs and outputs are read from top to bottom.

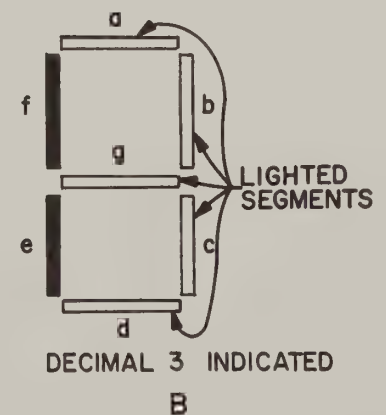
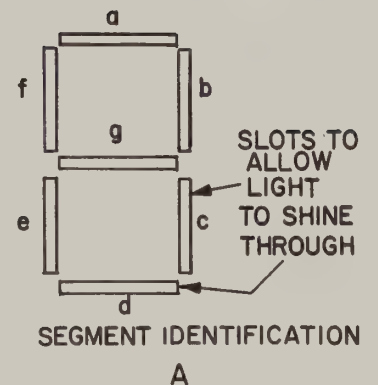


TWO FORMS OF THE EXCLUSIVE-OR FUNCTION

Figure 18

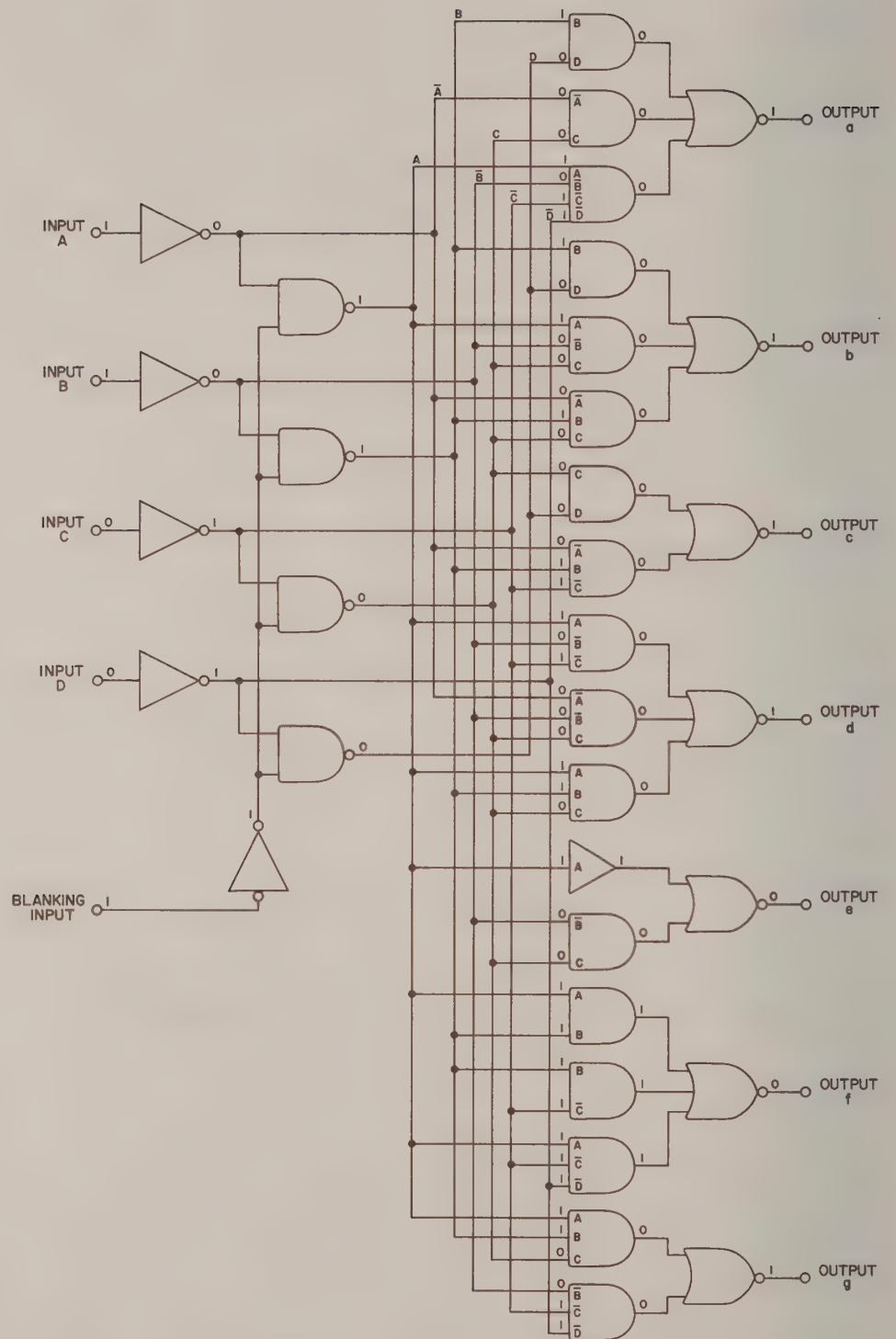
Two possible basic-logic arrangements for the EXCLUSIVE-OR function are shown in Figure 18. Either of these could replace each of the three functional gates in Figure 17. Other EXCLUSIVE-OR arrangements are also possible.

A type of decoder finding very wide use is the BCD-to-Seven-Segment decoder for use with seven indicators such as LED's (light-emitting diodes). The seven indicators are arranged as shown in Figure 19A. Under each slot is a LED or lamp (represented by letters a through g).



BCD-TO-SEVEN-SEGMENT DECODER DISPLAY

Figure 19



BCD-TO-SEVEN SEGMENT DECODER/DRIVERS

Figure 20

One decoding circuit is shown in Figure 20. Figure 20 shows the logic conditions for a decimal-3 input, a binary 0011 (where A is at 1, B is at 1, C is at 0, and D is at 0 and where A is the least significant bit). All of the outputs have logic 1's except e and f. Figure 19B shows how the number 3 is displayed when a BCD-to-Seven-Segment decoder has an input of 0011. Many times a BCD-to-Seven-Segment decoder is employed as a decoding driver, each output going to a base of a CE transistor whose collector is connected to 5, to 10 or even higher voltages through a lamp or LED. Thus, when an output terminal such as d has a logic 1, the transistor driven by that terminal is saturated, applying full supply voltage across the d lamp, causing it to illuminate.

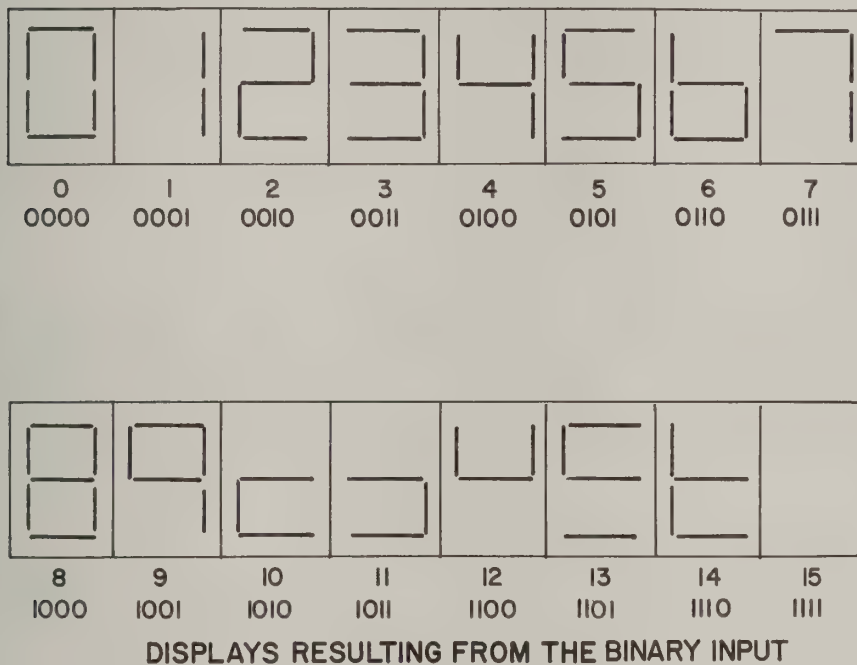


Figure 21

Figure 21 shows each of the possible displays for each decimal number input. In most cases binary numbers 1010 through 1111 are not allowed to enter the decoder, so that only decimal numbers 0 through 9 are displayed.

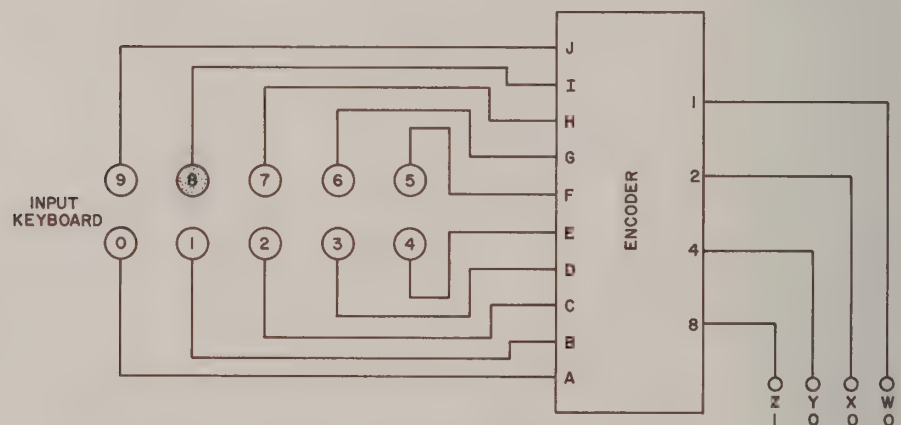
Note in Figure 20 that, for any lamps to be displayed, the blanking input must be at logic 1. This input is often used for lamp intensity control through the application of a variable pulse-width signal.

Some IC BCD-to-Seven-Segment decoders have the LEDS included in the package.

You can trace the logical digits through any type decoder in order to verify its operation. This ability will aid you in troubleshooting decoder arrangements. By knowing the correct binary state of any point in a decoder, you can check all points, and establish where the incorrect state is being generated.

ENCODING LOGIC

Just as it is necessary to convert binary information (used inside digital systems) to decimal information (for use outside the system), it is necessary to convert input decimal values into binary codes for entry into the system. For example, the input through a set of push buttons, representing the decimal digits 0 through 9, must be converted to a four-bit BCD code if this is used inside the computer.



DECIMAL-TO-BINARY ENCODING EXAMPLE

Figure 22

Figure 22 illustrates a typical computer input operation. When the key for decimal 8 is depressed, a 1-input occurs at input I of the encoder. Given this input, the encoder circuitry must generate the binary code group 1000. Table 16 shows the necessary truth data for this situation. From this, you can see that output W is 1 if input B or D or F or H or J is 1:

$$W = B + D + F + H + J.$$

Similarly:

$$X = C + D + G + H$$

$$Y = E + F + G + H$$

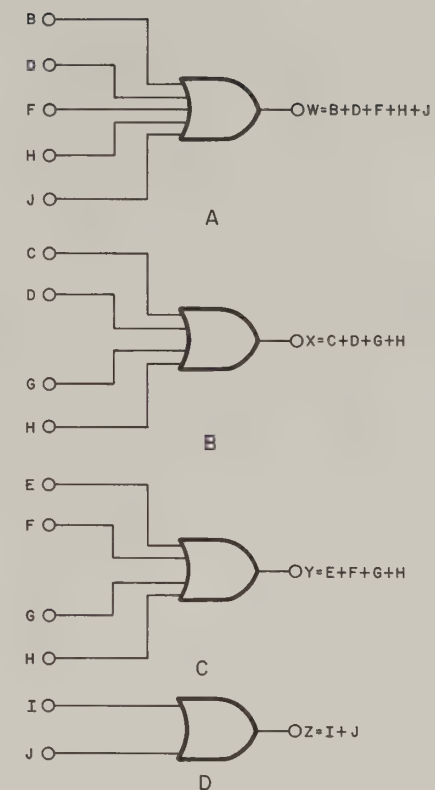
$$Z = I + J.$$

Decimal Number	Inputs										Outputs			
	J	I	H	G	F	E	D	C	B	A	Z	Y	X	W
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	0	0	1	0	0	0	0	1
2	0	0	0	0	0	0	0	1	0	0	0	0	1	0
3	0	0	0	0	0	0	1	0	0	0	0	0	1	1
4	0	0	0	0	0	1	0	0	0	0	0	1	0	0
5	0	0	0	0	1	0	0	0	0	0	0	1	0	1
6	0	0	0	1	0	0	0	0	0	0	0	1	1	0
7	0	0	1	0	0	0	0	0	0	0	0	1	1	1
8	0	1	0	0	0	0	0	0	0	0	1	0	0	0
9	1	0	0	0	0	0	0	0	0	0	1	0	0	1

Truth Data for Decimal-to-BCD Encoding

Table 16

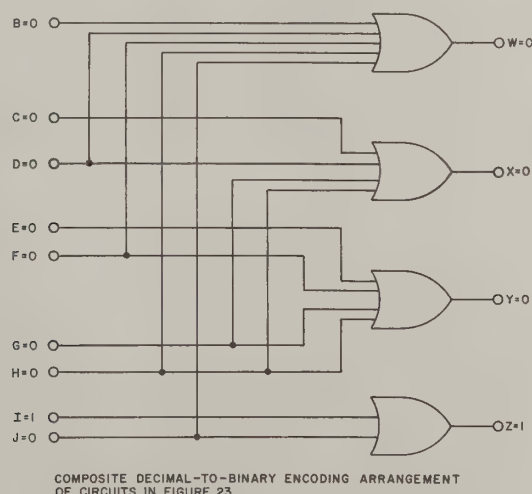
These are OR expressions. They can be implemented individually as shown in Figure 23. Note that input A in Figure 22 for the zero key is not used in Figure 23 or 24 because $W + X + Y + Z = 0$ when no other key is pressed. However, for a practical encoder, the inputs must be interconnected between the four OR gates. The result is shown in Figure 24. This figure also shows the verification for the output 1000 (read from bottom up) when input I occurs.



OR GATE IMPLEMENTATION OF OR FUNCTION OUTPUTS FOR ENCODING FUNCTION

Figure 23

Encoders are normally much easier to implement than decoders.



COMPOSITE DECIMAL-TO-BINARY ENCODING ARRANGEMENT OF CIRCUITS IN FIGURE 23

Figure 24

Decimal Number	Seven-Bit BCD Code	Code with Even Parity
0	0000000	00000000
1	0000001	10000001
2	0000010	10000010
3	0000011	00000011
4	0000100	10000100
5	0000101	00000101
6	0000110	00000110
7	0000111	10000111
8	0001000	10001000
9	0001001	00001001

Development of the Seven-Bit
BCD Code With Even Parity

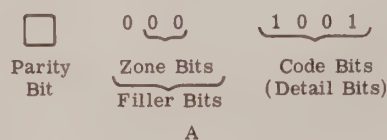
Table
17

PARITY CHECKING LOGIC

The coded information in a digital system should be free from error. However, due to electrical noise or component failure, it is possible for a code group to lose or gain an extra bit. One wrong bit might ruin a work piece being tooled by a computer-controlled machine. A simple and effective method of error checking called **PARITY CHECKING** may be used to guard against an error in a code group.

In parity checking, an extra bit called a **PARITY BIT** is added to each code group. The parity bit is chosen to obtain either an even number of 1's in all code groups, or an odd number of 1's in all code groups. Binary-coded decimal codes are the most suitable codes for parity checking. To illustrate how the parity bit is added, let's consider the BCD code.

Table 17 shows the BCD code with a parity bit added so that the total number of 1's in each code group of eight bits is even. That is, the final code has **EVEN PARITY**. Notice that the basic BCD code group shown in this table (middle column) uses seven bits rather than the four bits shown in Table 7. The extra three bits are "filler bits," as shown in Figure 25. Here, all the filler bits are 0's. They are used to enable the complete code group (with parity) to be an eight-bit code group. This will make the group compatible with the **PARITY DECODERS**. It is not uncommon to use filler bits in digital systems for reasons of compatibility. In some systems, special codes may



1 0 0 0 1 0 0 1
ODD PARITY (THREE 1's)
B

0 0 0 0 1 0 0 1
EVEN PARITY (TWO 1's)
C

EXPANSION OF FOUR-BIT CODE
GROUP TO SEVEN-BIT CODE GROUP
WITH PARITY BIT ADDED

Figure 25

The following Practice Exercise questions cover the subjects which you have just studied. They are:

DECODING LOGIC

AND EXPRESSION DECODING

PYRAMID DECODING

MATRIX DECODING

SPECIAL DECODER LOGIC

ENCODING LOGIC

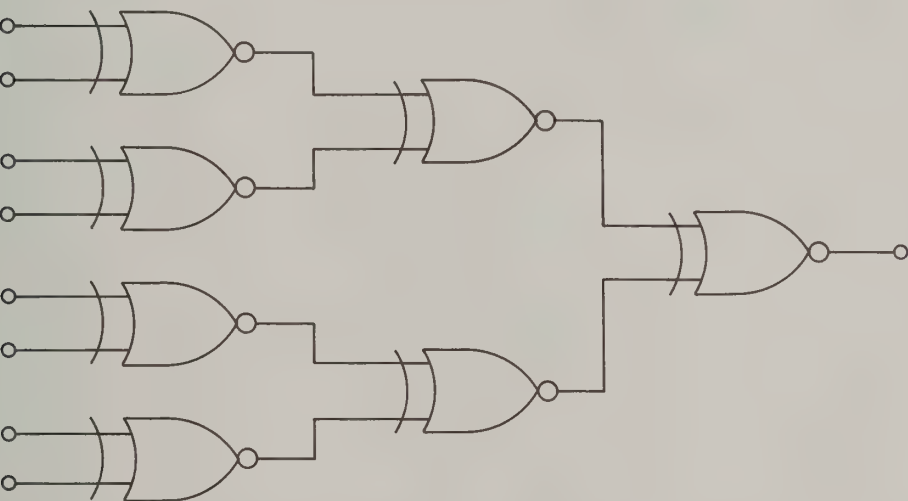
17. Does Equation 1 apply to the XS-3 code? If not, why not?
18. In the Gray code only one bit changes in progressing from one code group to the next. True or False?
19. What is the Gray code for the decimal number 35?
20. The Gray code is not a weighted code. True or False?
21. A decoder is a device used to convert binary-coded information into decimal information. True or False?
22. Draw a diagram showing that only Z_2 will provide a 1-output when the input to the decoder in Figure 9 is 10 (read from bottom to top).
23. Draw a diagram verifying the operation of the decoder in Figure 11 when the Gray code input is 100 (read from bottom to top).
24. List the AND expressions for the decoder in Figure 11.
25. A special five-bit code could be used to represent the 26 letters of the alphabet in binary-coded form. If five-input AND gates were used, how many AND gates are required to form an AND expression decoder to convert this code back into the alphabetical characters?
26. What is the most significant advantage of the pyramid decoder arrangement?

27. For a given-bit input code, the AND expression decoder requires more AND gates than a pyramid decoder. True or False?
28. If the code group 1011 is applied to the decoder in Figure 15, all outputs will be zero. True or False?
29. What advantage is associated with the matrix arrangement over the pyramid arrangement for performing the same decoding operation?

be entered into the filler positions. However, these codes would not interfere with the basic four-bit BCD code and would not have an effect on the required parity bit.

Table 18 shows the same BCD code with parity bits added so that the total number of 1's in each code group is odd. This code has **ODD PARITY**.

Although the use of a parity code means more bits are required to represent a given number, the advantage of built-in error checking is well worth the increase in bits. If an even parity code were used in a digital system, the system would check the even parity code for errors by noting that each code group contains an even number of 1's. If the code group did not have even parity, an error signal could be indicated. Similarly, if an odd parity code were used, the system would check for errors by noting that each code group contains an odd number of 1's. If not, an error signal could be indicated. In the event of an error, the code group can be rejected and retransmitted (manually or automatically) until it is received correctly.



EXCLUSIVE-NOR GATES ARRANGED AS AN EVEN PARITY DECODER

Figure 26

The even parity code groups in Table 17 can be checked by a logic arrangement such as is shown in Figure 26. Seven EXCLUSIVE-NOR gates are used. Each EXCLUSIVE-NOR gate has an associated truth table as shown in Table 19. This table may be used to follow the logical digits through the decoder (as shown in Figure 27) for both a correct and incorrect code group.

Decimal Number	Code with Odd Parity
0	10000000
1	00000001
2	00000010
3	10000011
4	00000100
5	10000101
6	10000110
7	00000111
8	00001000
9	10001001

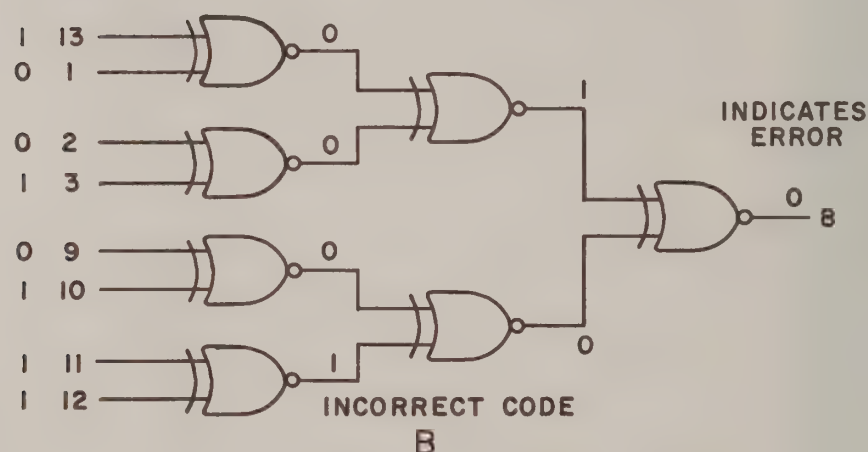
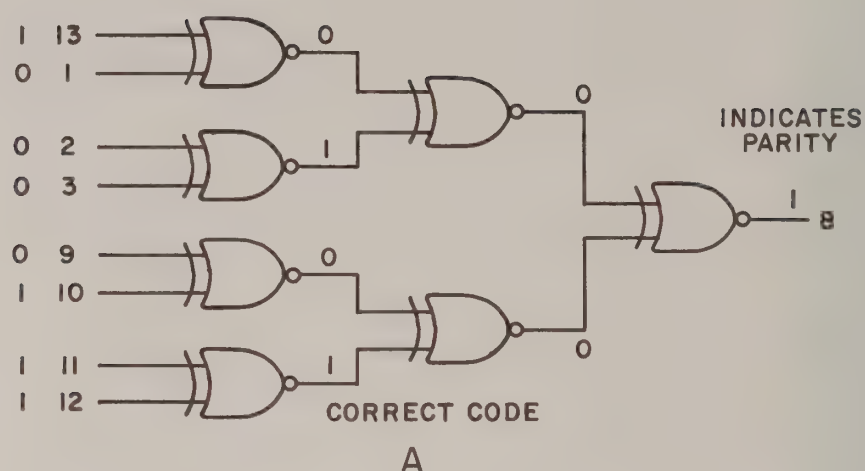
BCD Code With Odd Parity

Table 18

A	B	Z
1	1	1
1	0	0
0	1	0
0	0	1

Truth Data For EXCLUSIVE-NOR

Table 19

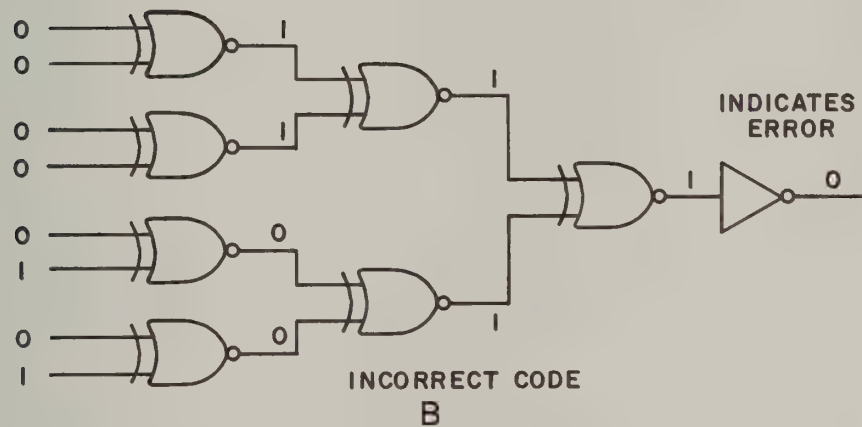
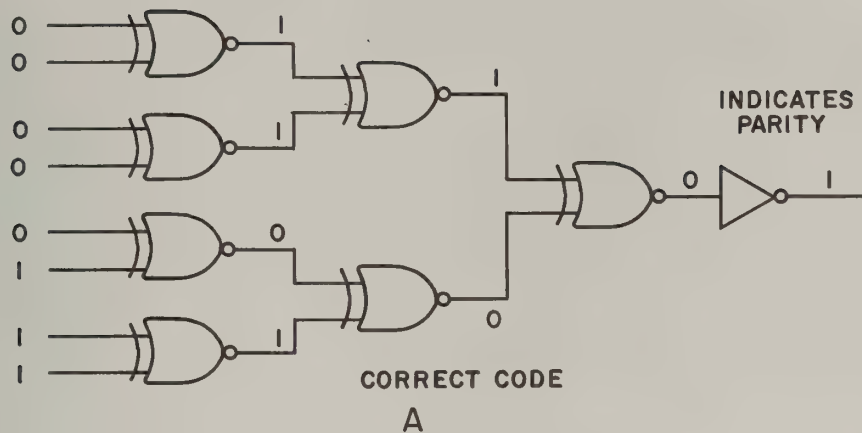


OPERATION OF EVEN PARITY DECODER

Figure 27

The decoder in Figure 26 can be converted to check for odd parity by adding an inverted (NOT) element at the output (Figure 28). A correct and an incorrect code group check is shown in the figure.

Other error-coding techniques are also available. Parity checking is the simplest. However, it is possible for two errors to occur, one offsetting the effects of the other. In this case, parity would check, but the code group would be erroneous. More complex codes are used to overcome this problem when multiple errors have a high chance of occurring.



CONVERSION OF EVEN PARITY DECODER
TO ODD PARITY DECODER WITH
EXAMPLE OPERATIONS

Figure 28

COMPUTER CODES

As mentioned earlier, a digital system works with information which is in some form of binary code. Actually, the system works with electric signals representing the coded information. If a computing system is primarily designed to perform a large number of computations, it would use a code such as the natural binary code or maybe the XS-3 code. A computing system used to keep track of the inventory of some company might use some form of the BCD code. Most digital computers incorporate some error-checking mechanism such as a parity bit in their code.

Alphanumeric Coding

Many digital systems handle only numbers. For example, computers designed for scientific applications are only required to solve problems. However, a large number of digital systems, such as the data-processing systems used in the business world, have to handle letters as well as numbers. People, organizations, and goods all have names and thus require the use of alphabetical characters. In addition, the data-processing systems must use numbers to represent addresses, prices, and quantities. When a code represents both numbers and letters, it is called an ALPHANUMERIC or ALPHAMERIC CODE.

A very simple method of coding both numbers and letters is to use a two-digit decimal number to represent each letter. For example, the letter A may be represented by 51, the letter B by 52, and so on. Any one of the codes previously described may then be used to represent the number. For example, assume the BCD code is being used. If A is represented by 51, then A is 0101 0001 (51 is 0101 0001 in the BCD code).

This system has one disadvantage. When numbers are to be used, they must also be represented by a two-digit decimal number. As an example, 1 may be represented by the decimal number 41, 2 by 42, and so on. Then, one of the codes previously described can be used to represent the two-digit decimal number.

Another method of alphanumeric coding is to use the XS-3 code and add extra bits to enable the code groups to represent numbers as well as letters. Such a coding arrangement based on the XS-3 code is shown in Table 20. Six bits make up each code group. The first two bits to the left of each code group are called the ZONE BITS. They determine whether the group of bits is representing a number or a letter. The next four bits are often called the DETAIL BITS. Also see Figure 25 about zone and detail bits. Detail bits determine the exact number or letter the entire code group represents.

Notice in Table 20 that the numerical value of the code group representing the letter A is less than that representing B. Likewise, the numerical value of the code group representing Z is greater than that of any other letter. This order provides a means by which a digital system can arrange information in alphabetical order. All the system has to do is arrange the information in numerical order. For example, a group of persons' names may be put in alphabetical order by placing the code groups which represent each name in numerical order.

CHARACTER	ALPHA- NUMERIC	CHARACTER	ALPHA- NUMERIC	CHARACTER	ALPHA- NUMERIC	CHARACTER	ALPHA- NUMERIC
	00 0000 (0)		*01 0000 (10)		*10 0000 (20)		*11 0000 (30)
	00 0001 (1)	,	*01 0001 (11)		*10 0001 (21)		*11 0001 (31)
-	00 0010 (2)	.	*01 0010 (12)		*10 0010 (22)		*11 0010 (32)
0	00 0011 (3)	;	*01 0011 (13)	/	10 0011 (23)		11 0011 (33)
1	00 0100 (4)	A	01 0100 (14)	J	10 0100 (24)	+	11 0100 (34)
2	00 0101 (5)	B	01 0101 (15)	K	10 0101 (25)	S	11 0101 (35)
3	00 0110 (6)	C	01 0110 (16)	L	10 0110 (26)	T	11 0110 (36)
4	00 0111 (7)	D	01 0111 (17)	M	10 0111 (27)	U	11 0111 (37)
5	00 1000 (8)	E	01 1000 (18)	N	10 1000 (28)	V	11 1000 (38)
6	00 1001 (9)	F	01 1001 (19)	O	10 1001 (29)	W	11 1001 (39)
7	00 1010 (10)	G	*01 1010 (20)	P	10 1010 (30)	X	11 1010 (40)
8	00 1011 (11)	H /	01 1011 (21)	Q	10 1011 (31)	Y	11 1011 (41)
9	00 1100 (12)	I /	01 1100 (22)	R	10 1100 (32)	Z	11 1100 (42)
	00 1101 (13)		*01 1101 (23)		*10 1101 (33)		11 1101 (43)
	*00 1110 (14)		*01 1110 (24)		*10 1110 (34)		11 1110 (44)
	*00 1111 (15)		*01 1111 (25)		*10 1111 (35)		11 1111 (45)

*Alphanumeric Values not used for sorting because of duplication

An Alphanumeric Code Based on Zoning and the Use of the XS-3 Code

Table 20

There are no symbols shown for some of the code groups in Table 20. These code groups can be used for special instructions for the system. For example, the code group 00 0001 might be used to tell the digital system to stop.

Specific Computer Codes

A computer code that is based on the BCD code appears in Table 21. This code is used in the IBM 705 computing system. With this code, every character is represented by a seven-bit code group. As shown at the top of the table, each seven-bit code group is composed of three parts; a parity bit, two zone bits and four detail bits. The detail bits are the BCD codes for the decimal numbers 0 to 9, plus additional code groups which represent symbols

CHARACTER	PARITY	ZONE	DETAIL	CHARACTER	PARITY	ZONE	DETAIL	CHARACTER	PARITY	ZONE	DETAIL	CHARACTER	PARITY	ZONE	DETAIL
Drum Mark	0	00	0000	Blank	1	01	0000	-	1	10	0000	&	0	11	0000
1	1	00	0001	/	0	01	0001	J	0	10	0001	A	1	11	0001
2	1	00	0010	S	0	01	0010	K	0	10	0010	B	1	11	0010
3	0	00	0011	T	1	01	0011	L	1	10	0011	C	0	11	0011
4	1	00	0100	U	0	01	0100	M	0	10	0100	D	1	11	0100
5	0	00	0101	V	1	01	0101	N	1	10	0101	E	0	11	0101
6	0	00	0110	W	1	01	0110	O	1	10	0110	F	0	11	0110
7	1	00	0111	X	0	01	0111	P	0	10	0111	G	1	11	0111
8	1	00	1000	Y	0	01	1000	Q	0	10	1000	H	1	11	1000
9	0	00	1001	Z	1	01	1001	R	1	10	1001	I	0	11	1001
0	0	00	1010	≠	1	01	1010	ō	1	10	1010	ō	0	11	1010
#	1	00	1011	,	0	01	1011	\$	0	10	1011	.	1	11	1011
@	0	00	1100	%	1	01	1100	*	1	10	1100	□	0	11	1100
N. U.	1	00	1101	N. U.	0	01	1101	N. U.	0	10	1101	N. U.	1	11	1101
N. U.	1	00	1110	N. U.	0	01	1110	N. U.	0	10	1110	N. U.	1	11	1110
Tape Mark	0	00	1111	N. U.	1	01	1111	N. U.	1	10	1111	≠	0	11	1111

The Alphanumeric Code Used in the IBM 705 Computing System

Table 21

such as #, @, and others. The zone bits are added to the detail bits to enable the letters of the alphabet and special symbols, as well as numbers, to be represented. The zone bits determine whether the code group represents a number, letter, or symbol.

Besides the commonly used numbers, letters and symbols, the code includes some special ones. For example, the Tape Mark and Drum Mark are special instructions which are used by the computing system. The code groups indicated as N. U. are Not Used by the computing system.

The parity bit is added to each code group so that every code group contains an even number of 1's.

The computer code of Table 21 is an alphanumeric code. Provision for numbers and letters is provided. This code is similar to the alphanumeric code of Table 20.

A computer code similar to the one used in Table 21 is shown in Table 22. This code is also a seven-bit code, and it is based on the XS-3 code. The

CHARACTER	PARITY	ZONE	DETAIL	CHARACTER	PARITY	ZONE	DETAIL	CHARACTER	PARITY	ZONE	DETAIL	CHARACTER	PARITY	ZONE	DETAIL
i	1	00	0000	r	0	01	0000	t	0	10	0000	Σ	1	11	0000
Δ	0	00	0001	,	1	01	0001	"	0	10	0001	β	0	11	0001
-	0	00	0010	.	1	01	0010	/	1	10	0010	:	0	11	0010
0	1	00	0011	;	0	01	0011	)	0	10	0011	+	1	11	0011
1	0	00	0100	A	1	01	0100	J	1	10	0100	/	0	11	0100
2	1	00	0101	B	0	01	0101	K	0	10	0101	S	1	11	0101
3	1	00	0110	C	0	01	0110	L	0	10	0110	T	1	11	0110
4	0	00	0111	D	1	01	0111	M	1	10	0111	U	0	11	0111
5	0	00	1000	E	1	01	1000	N	1	10	1000	V	0	11	1000
6	1	00	1001	F	0	01	1001	O	0	10	1001	W	1	11	1001
7	1	00	1010	G	0	01	1010	P	0	10	1010	X	1	11	1010
8	0	00	1011	H	1	01	1011	Q	1	10	1011	Y	0	11	1011
9	1	00	1100	I	0	01	1100	R	0	10	1100	Z	1	11	1100
'	0	00	1101	#	1	01	1101	\$	1	10	1101	%	0	11	1101
&	0	00	1110	¢	1	01	1110	*	1	10	1110	=	0	11	1110
(	1	00	1111	@	0	01	1111	?	0	10	1111	N.U.	1	11	1111

The Alphanumeric Code used in the Sperry Rand Univac II Computing System

Table 22

code of Table 22 is also composed of three parts: a parity bit, two zone bits and four detail bits. The detail bits are the XS-3 codes for the decimal numbers 0 to 9, with additional groups for various symbols. Two zone bits are added to the detail bits to enable the representation of the letters of the alphabet and special symbols, as well as numbers. The parity bit is added as a means of error checking. It is adjusted so that each code group contains an odd number of 1's (odd parity).

By using the XS-3 code, arithmetic operations are easily performed. The code of Table 22 is used by the Sperry Rand Corporation in their Univac II computing system.

A computer code which is quite different from those shown in Tables 21 and 22 appears in Table 23. This code is an eight-bit code with every number, letter, and symbol (except the + and - symbols) represented by an eight-bit code group. This code uses a very simple coding arrangement.

CODING AND DECODING CIRCUITS

CHARACTER		DECIMAL FORM	CHARACTER		DECIMAL FORM
A	0100 0001	41	BLANK	0000 0000	00
B	0100 0010	42	.	0000 0011	03
C	0100 0011	43		0000 0100	04
D	0100 0100	44	&	0001 0000	10
E	0100 0101	45	\$	0001 0011	13
F	0100 0110	46	*	0001 0100	14
G	0100 0111	47	-	0010 0000	20
H	0100 1000	48	/	0010 0001	21
I	0100 1001	49	,	0010 0011	23
J	0101 0001	51	%	0010 0100	24
K	0101 0010	52	#	0011 0011	33
L	0101 0011	53	@	0011 0100	34
M	0101 0100	54	+0	0100 0000	40
N	0101 0101	55	-0	0101 0000	50
O	0101 0110	56	0	1000 0000	80
P	0101 0111	57	1	1000 0001	81
Q	0101 1000	58	2	1000 0010	82
R	0101 1001	59	3	1000 0011	83
S	0110 0010	62	4	1000 0100	84
T	0110 0011	63	5	1000 0101	85
U	0110 0100	64	6	1000 0110	86
V	0110 0101	65	7	1000 0111	87
W	0110 0110	66	8	1000 1000	88
X	0110 0111	67	9	1000 1001	89
Y	0110 1000	68	+	0001 0000 1001	109
Z	0110 1001	69	-	0000 0000 1001	009

A Computer Code based on the Assignment of Decimal Numbers to Each Character

Table 23

Every number, letter, and symbol (except the + and - symbols) is first represented by a two-digit decimal number. Then, this two-digit decimal number is encoded into the BCD code.

For example, the letter A (with this code) is represented by the decimal number 41. Using Table 8, the BCD code for 41 is 0100 0001, and this is the eight-bit code for A. As another example, consider the percent symbol (%). This symbol is represented by the decimal number 24. Again from Table 7, the BCD code for the number 24 is 0010 0100, which is the code for the % symbol.

Table 23 shows the decimal numbers which represent each one of the letters, numbers, and symbols before the eight-bit code is formed. This code is also an alphanumeric code; however, provisions for error checking are not provided in this code. If desired, a parity bit could be added to enable each code group to contain an odd number of 1's if an odd parity is desired, or an even number of 1's if an even parity is desired.

SUMMARY

Most digital systems work with information which is in the form of some binary code. The binary code used in a particular digital system is called the machine language of the system.

The binary number system is, in itself, a code. When used as a code, it is called the natural binary code. It is the most efficient code since it uses the least number of bits to represent a given decimal number. In addition, it serves as the building block for other codes.

In the binary number system only two symbols are used, 0 and 1. These symbols are arranged in a manner similar to the ten symbols used in the more common decimal number system. That is, in both systems, the symbols are arranged in a positional relationship to represent different powers of the number system radix. In the decimal system, the radix is 10; in the binary system, the radix is 2. Fractional values are represented by negative powers of the system radix. Binary numbers can be converted to equivalent decimal numbers by using this relationship. Decimal numbers are converted to equivalent binary numbers with the aid of a positional table.

A binary-coded decimal (BCD) code can be derived from the natural binary code. With the BCD code each digit of the decimal number being coded is represented by a group of four bits. These four bits are the natural binary code of the digit being represented. A code quite similar to the BCD

code is the XS-3 code. As in the BCD code, each digit of the decimal number being coded is represented with four bits. The XS-3 BCD code is actually the BCD code of each digit being coded plus 3.

A code quite different from the XS-3 and BCD codes is the Gray code. In this code only one bit changes in progressing from one number to the next. For this reason it is often called the cyclic code.

In order to accept decimal information from a human operator and then present the operator with decimal results, digital systems must contain encoding and decoding logic. Encoding logic can be developed on the basis of simple OR gate combinations, provided the inputs are made through a single-pulse type entry (such as a keyboard). Decoders may be developed on the basis of AND gates for each code conversion maxterm expression; arranged in pyramid fashion, taking advantage of common maxterm components; or arranged in a matrix. The matrix is preferred with diode logic; the pyramid, with the various transistor logic circuits. Decoders may also be developed through the use of more complex logic functions.

Codes can be made self-checking by the addition of a parity bit. Depending on the arrangement used, the parity bit makes each code group contain an even or odd number of 1's.

A code which represents both numbers and letters is called an alphanumeric code. Codes such as the XS-3 and the BCD code are converted to alphanumeric codes by the addition of zone bits. Other codes use assigned decimal numbers to represent each character. These decimal numbers are then converted to one of the binary codes. Computer systems may use any one of several possible alphanumeric codes.

The following Practice Exercise questions cover the subjects which you have just studied. They are:

PARITY CHECKING LOGIC

COMPUTER CODES

ALPHANUMERIC CODING

SPECIFIC COMPUTER CODES

30. Diode gates are generally used in the pyramid decoder; transistor gates, in the matrix decoder. True or False?
31. Analyze the circuit of Figure 20 for a decimal 8 (binary 1000) input and draw a figure similar to Figure 19A showing which lamps would be on.
32. Decimal-to-binary encoding can be accomplished with simple OR gate arrangements if the data entry is made with punch keys. True or False?
33. Verify that the output of Figure 24 is the code group 0110 when the input is G (the decimal 6 key is punched in Figure 22).
34. The table below provides the encoding truth data for converting decimal digits into XS-3 code groups. Write the four OR encoding expressions and sketch an encoding arrangement using four OR gates to implement the conversion.

Decimal Number	Inputs										Outputs			
	J	I	H	G	F	E	D	C	B	A	Z	Y	X	W
0	0	0	0	0	0	0	0	0	0	1	0	0	1	1
1	0	0	0	0	0	0	0	0	1	0	0	1	0	0
2	0	0	0	0	0	0	0	1	0	0	0	1	0	1
3	0	0	0	0	0	0	1	0	0	0	0	1	1	0
4	0	0	0	0	0	1	0	0	0	0	0	1	1	1
5	0	0	0	0	1	0	0	0	0	0	1	0	0	0
6	0	0	0	1	0	0	0	0	0	0	1	0	0	1
7	0	0	1	0	0	0	0	0	0	0	1	0	1	0
8	0	1	0	0	0	0	0	0	0	0	1	0	1	1
9	1	0	0	0	0	0	0	0	0	0	1	1	0	0

Truth Data for XS-3 Code Encoding

35. Parity is used as a _____ against code group errors.
36. An odd parity code always has an _____ number of 1's in each code group.

37. Verify that parity checks with the decoder of Figure 26 when the input code group is 10000100.
38. Verify that parity does not check with the decoder of Figure 28 when the input code group is 10000010.
39. A seven-position code group for the Gray code equivalent of decimal 9 is 0001101. Add the parity bit required to produce a code group with odd parity.
40. In an alphanumeric code similar to the one shown in Table 20, the detail bits determine whether the code group represents a number or a letter. True or False?
41. Use the numerical value of the code groups in Table 20 to show how, on the basis of total numerical value, the words CAN and CAT can be placed in alphabetical order by a digital computer.

IMPORTANT DEFINITIONS

ALPHAMERIC CODE—ALPHANUMERIC CODE.

ALPHANUMERIC CODE—A code which has provisions for representing both numbers and letters.

AND EXPRESSION DECODER—A DECODING CIRCUIT which uses one AND gate for each term in the decoding truth table.

BINARY CODE—Any code composed of only 0's and 1's.

BINARY-CODED DECIMAL (BCD) CODE—A code based on the NATURAL BINARY CODE. Each digit of the number being coded is represented by a group of four bits which is the natural binary codes of the digit being coded.

BIT—A contraction of Binary Digit.

CODE—A set of symbols used to represent information.

CYCLIC CODE—A binary code which changes only one bit when advancing from one number to the next.

DECIMAL NUMBER SYSTEM—A numerical coding arrangement consisting of the digits 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9.

DECODING CIRCUITS—Logic arrangements used to convert from a binary code to the decimal system, or from one binary code to another.

DETAIL BITS—That portion of an ALPHANUMERIC CODE which actually represents the number, letter, or character being represented.

ENCODING CIRCUITS—Logic arrangements used to convert from the decimal system to one of the binary codes.

EVEN PARITY—A code which contains an even number of 1's in each code group.

EXCESS-3 (XS-3) CODE—A code based on the NATURAL BINARY CODE. The XS-3 code is the BCD code of the digit being coded plus 3. It is not a weighted code.

FORBIDDEN COMBINATIONS—Code group combinations which can be formed, but which have no meaning in a given code.

GRAY CODE—A code formed from the NATURAL BINARY CODE. The Gray code is often called a CYCLIC CODE because it only changes one bit in going from one number to the next succeeding one. It is not a weighted code.

IMPORTANT DEFINITIONS (Continued)

MATRIX DECODER—A DECODING CIRCUIT whose logic elements, inputs, and outputs are arranged in a column and row interconnection.

NATURAL BINARY CODE—A code based on the binary number system.

ODD PARITY—A code which contains an odd number of 1's in each code group.

PARITY BIT—A BIT added to a code to make the code self-checking. The parity bit may be adjusted to enable each code group to contain an even or odd number of 1's.

PARITY CHECKING—A process of checking a code group for an error on the basis of EVEN or ODD PARITY.

PARITY DECODER—Logic arrangements for performing the PARITY CHECKING operation.

PYRAMID DECODER—A DECODING CIRCUIT which uses AND gates arranged in a pyramid fashion, each one feeding two others. This takes advantage of common maxterm components when several inputs are involved.

RADIX—The number of symbols used in a number system.

WEIGHT—The multiplier of a particular digit in a given number system. It is represented by R^n (whole number powers of the radix).

ZONE BITS—That portion of an ALPHANUMERIC CODE which identifies whether the code represents a number, a letter, or some other character.

ESSENTIAL SYMBOLS AND EQUATIONS

d_3, d_2, d_1, d_0 The specific digits in any numerical system, representing a whole number

d_{-1}, d_{-2}, d_{-3} The specific digits in any numerical system, representing a fractional number

N A decimal number

R The radix of any numerical system

$$N = \dots + d_3R^3 + d_2R^2 + d_1R^1 + d_0R^0 \quad (1)$$

$$N = d_{-1}R^{-1} + d_{-2}R^{-2} + d_{-3}R^{-3} + \dots \quad (2)$$

PRACTICE EXERCISES SOLUTIONS

1. four

2. 10^6

$$3. N = 7(10)^3 + 8(10)^2 + 4(10)^1 + 2(10)^0 = 7000 + 800 + 40 + 2.$$

$$4. N = 6(10)^{-1} + 5(10)^{-2} = 0.6 + 0.05$$

$$5. N = 6(10)^2 + 9(10)^1 + 1(10)^0 + 2(10)^{-1} + 8(10)^{-2} \\ = 600 + 90 + 1 + 0.2 + 0.08$$

$$6. (a) 1(2)^8 + 0(2)^7 + 1(2)^6 + 0(2)^5 + 0(2)^4 + 1(2)^3 + 0(2)^2 + 0(2)^1 + 1(2)^0 \\ = 256 + 64 + 8 + 1 = 329.$$

$$(b) 1(2)^7 + 0(2)^6 + 1(2)^5 + 0(2)^4 + 1(2)^3 + 0(2)^2 + 1(2)^1 + 0(2)^0 \\ = 128 + 32 + 8 + 2 = 170.$$

$$(c) 1(2)^5 + 1(2)^4 + 0(2)^3 + 0(2)^2 + 0(2)^1 + 1(2)^0 = 32 + 16 + 1 = 49.$$

$$(d) 1(2)^5 + 1(2)^4 + 0(2)^3 + 1(2)^2 + 0(2)^1 + 1(2)^0 = 32 + 16 + 4 + 1 = 53.$$

$$(e) 1(2)^6 + 0(2)^5 + 0(2)^4 + 0(2)^3 + 0(2)^2 + 0(2)^1 + 0(2)^0 \\ = 64 + 0 + 0 + 0 + 0 = 64.$$

$$(f) 1(2)^3 + 0(2)^2 + 1(2)^1 + 1(2)^0 + 0(2)^{-1} + 0(2)^{-2} + 1(2)^{-3} \\ = 8 + 2 + 1 + 1/8 \\ = 11\frac{1}{8}$$

7. (a)

32	16	8	4	2	1
1	1	0	1	0	1

$$53 - 32 = 21$$

$$21 - 16 = 5$$

$$5 - 4 = 1$$

$$1 - 1 = 0$$

(b)

64	32	16	8	4	2	1
1	0	0	0	0	0	0

$$64 - 64 = 0$$

(c)

64	32	16	8	4	2	1
1	0	0	0	1	1	1

$$71 - 64 = 7$$

$$7 - 4 = 3$$

$$3 - 2 = 1$$

$$1 - 1 = 0$$

PRACTICE EXERCISE SOLUTIONS (Continued)

(d)

64	32	16	8	4	2	1
1	1	0	0	1	1	0

$$102 - 64 = 38$$

$$38 - 32 = 6$$

$$6 - 4 = 2$$

$$2 - 2 = 0$$

(e)

4	2	1		$\frac{1}{2}$
1	0	1	.	1

$$5.5 - 4 = 1.5$$

$$1.5 - 1 = .5$$

$$.5 - .5 = 0$$

(f)

8	4	2	1		$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$
1	1	0	0	.	0	1	1

$$12.375 - 8 = 4.375$$

$$4.375 - 4 = .375$$

$$.375 - .25 = .125$$

$$.125 - .125 = 0$$

8. Weights are powers of the radix.

9. Using the method of Practice Exercise 7, you should have:

32	16	8	4	2	1
1	1	1	0	0	1

The binary number for decimal 57 is 111001; then the weights are 1, 8, 16, and 32.

10. False—Each digit is a four-bit code group.

11. The BCD codes for the decimal numbers 1, 9, 7 and 5 from Table 7 are 0001, 1001, 0111 and 0101, respectively. Therefore, the BCD code for the decimal number 1975 is 0001 1001 0111 0101.

12. From Table 7 the codes 1000, 0110, 1001 represent the decimal numbers 8, 6 and 9, respectively. Thus, the BCD code 1000 0110 1001 represents the decimal number 869.

13. False—In the BCD code 1101 is a forbidden combination. The decimal number 13 is 0001 0011 in this code.

PRACTICE EXERCISE SOLUTIONS (Continued)

14. True

15. From Table 8, the XS-3 codes for 1, 7 and 3 are 0100, 1010 and 0110, respectively. Therefore, the XS-3 code for 173 is 0100 1010 0110.

16. From Table 8 the code groups 1010, 1100 and 0110 represent 7, 9 and 3, respectively. Therefore, 1010 1100 0110 represents the decimal number 793.

17. No—Decimal code zero is not zero in the XS-3 code.

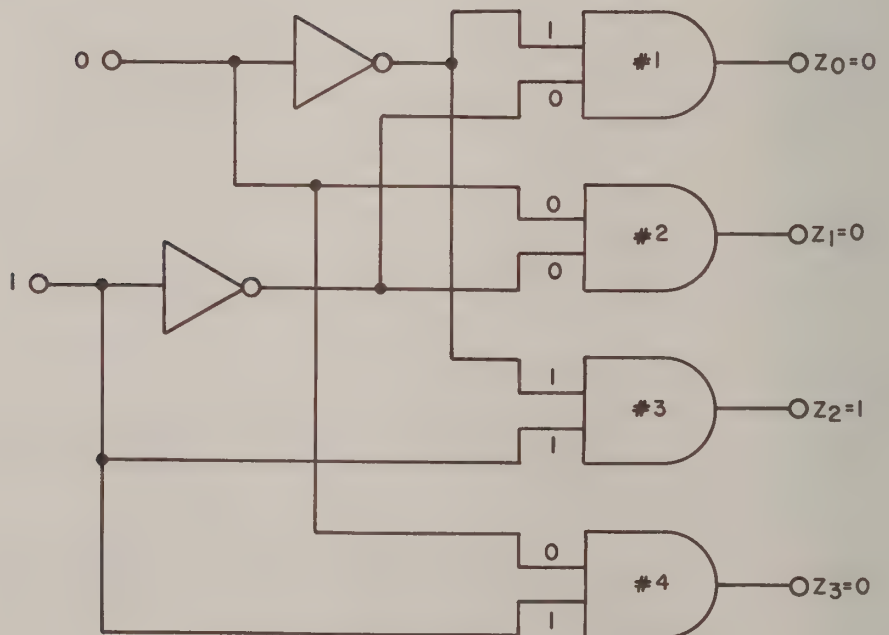
18. True

19. 110010—100011 Natural Binary Code for 35
100011 Shift and Disregard Right-most Digit
 110010 Gray Code for 35

20. True

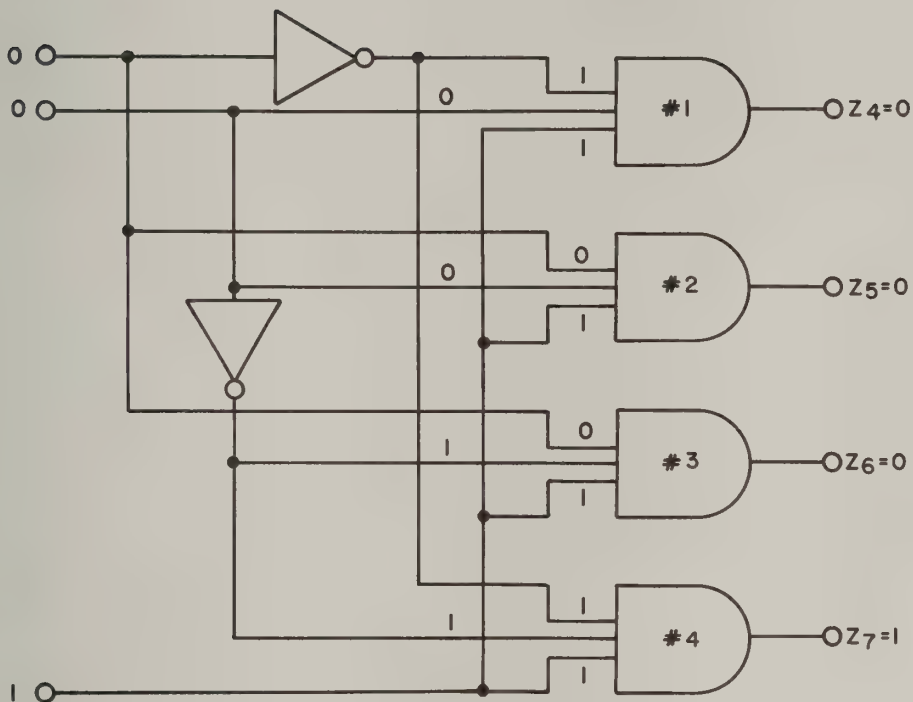
21. True

22.



PRACTICE EXERCISE SOLUTIONS (Continued)

23.



$$24. Z_4 = \bar{A} \cdot B \cdot C$$

$$Z_5 = A \cdot B \cdot C$$

$$Z_6 = A \cdot \bar{B} \cdot C$$

$$Z_7 = \bar{A} \cdot \bar{B} \cdot C$$

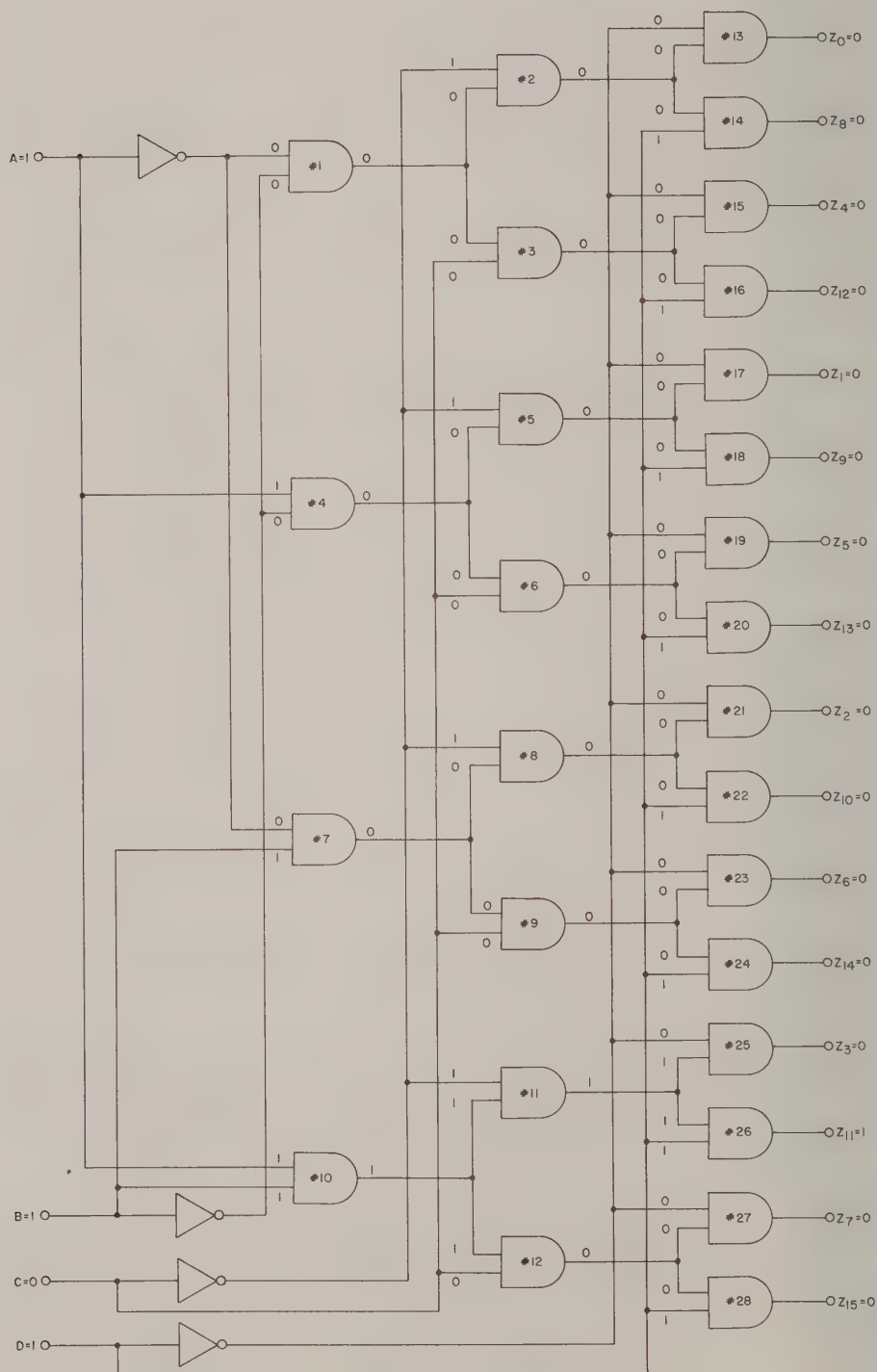
25. 26—One AND gate for each output character.

26. The most significant advantage of the pyramid decoder arrangement is the fact that it can be expanded to accept larger code groups simply by adding a new level of pyramided AND gates, using the existing outputs.

27. False—With the proper arrangement, an AND expression decoder might use fewer gates than the pyramid decoder.

PRACTICE EXERCISE SOLUTIONS (Continued)

28. False— Z_{11} has a logical output of 1.

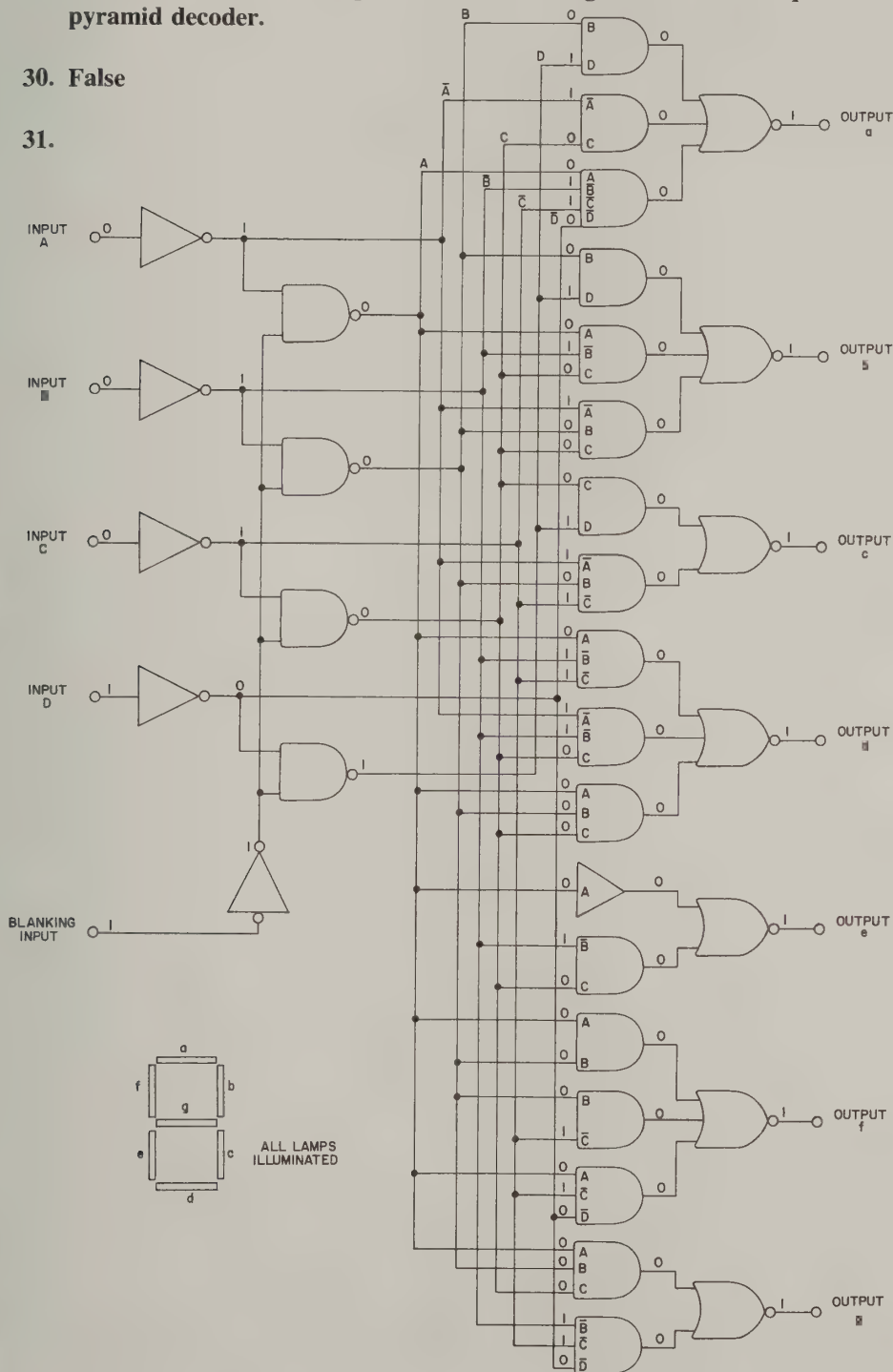


PRACTICE EXERCISE SOLUTIONS (Continued)

29. The matrix decoder requires fewer AND gates than the equivalent pyramid decoder.

30. False

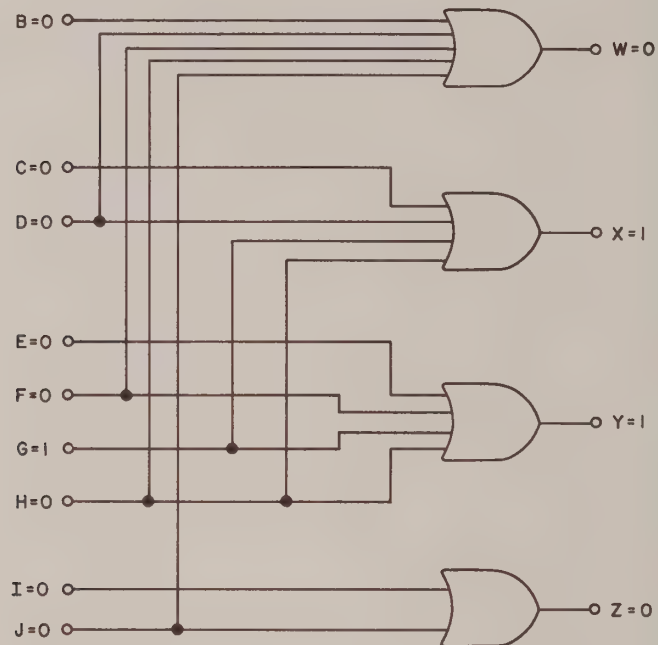
31.



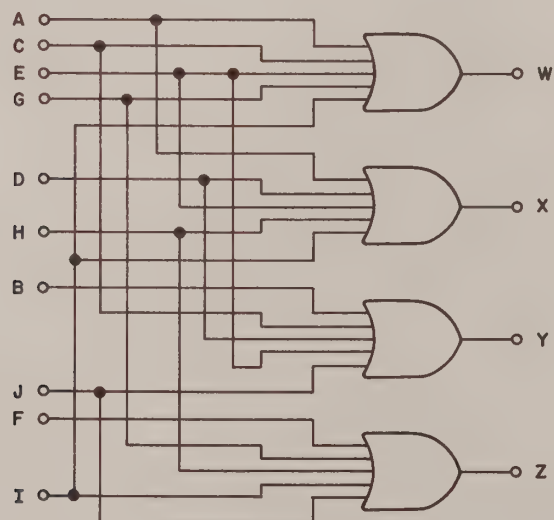
PRACTICE EXERCISE SOLUTIONS (Continued)

32. True

33.



34. $W = A + C + E + G + I$
 $X = A + D + E + H + I$
 $Y = B + C + D + E + J$
 $Z = F + G + H + I + J$

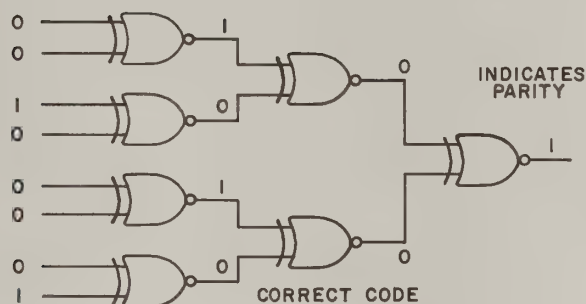


PRACTICE EXERCISE SOLUTIONS (Continued)

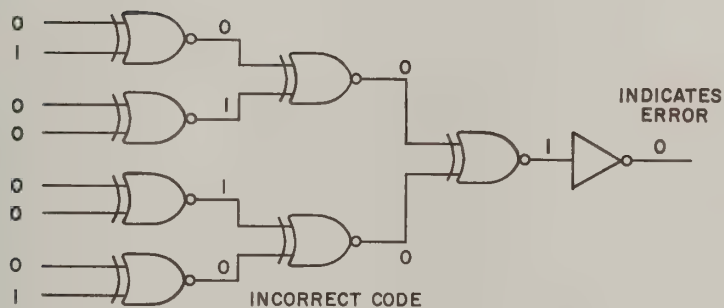
35. check or guard

36. odd

37.



38.



39. 00001101 (The parity bit is 0.)

40. False—The zone bits provide this determination.

41. The decimal number values for each letter are:

$$C = 01\ 0110 = 16$$

$$A = 01\ 0100 = 14$$

$$N = 10\ 1000 = 28$$

$$T = 11\ 0110 = 36$$

From this, the total values for the words are:

$$CAN = 16 + 14 + 28 = 58$$

$$CAT = 16 + 14 + 36 = 66$$

Therefore, when the words are placed in numerical order, they are also in alphabetical order.



ONE OF THE

BELL & HOWELL SCHOOLS

1. D - ACCORDING TO EQUATION 1 OF THIS LESSON, THE ACTUAL VALUE OF THE 3 IN THE DECIMAL NUMBER 6,731 IS -- 30.

Equation 1 states $N = d_3R^3 + d_2R^2 + d_1R^1 + d_0R^0$, where N is the number, R is the radix and d is the digit of the number. Since 3 in the decimal number 6,731 is in the form of d_1R^1 , its actual value will be $3(10)^1$ or 30. The digit is 3, the radix is 10 and 10^1 is 10.

2. C - THE BINARY NOTATION FOR THE DECIMAL NUMBER 73 IS -- 1001001.

The decimal-to-binary conversion can be performed using a table similar to the one shown in Figure 2. The first step in this conversion is to note the highest number (or weight) in the table which is equal to or less than 73. The highest weight in the table which is equal to or less than 73 is 64. Place a 1 in the block under 64 and subtract 64 from 73 to give 9. Now note the highest weight in the table which is equal to or less than 9. The highest weight in the table which is equal to or less than 9 is 8. Therefore, place a 1 in the block under 8 and subtract 8 from 9 to give 1. The highest number equal to 1 is 1. Therefore, place a 1 in the block under 1. The binary notation for the decimal number is now extracted from the table as shown:

256	128	64	32	16	8	4	2	1
0	0	1	0	0	1	0	0	1

0001001

3. D - THE DECIMAL NOTATION FOR THE BINARY NUMBER 1011001 IS -- 89.

The binary-to-decimal conversion can also be performed using a table similar to the one shown in Figure 2. First, place the binary number to be converted into the table. (Note: The right-most digit must be placed under 1 and the following digits be placed under the appropriate column in the table.) Next, add up all the decimal numbers which have 1's under them as shown below:

256	128	64	32	16	8	4	2	1
0	0	1	0	1	1	0	0	1

64
16
8
4
1
89

4. D - THE DECIMAL NOTATION FOR THE BINARY NUMBER 1011101.01 IS -- 93.25.

The binary-to-decimal conversion for fractional numbers can be performed using a table similar to the one shown in Figure 4. First, place the binary number to be converted into the table. Place the binary digits under the appropriate column. Next, add up all the decimal numbers which have 1's under them as shown below.

64	32	26	8	4	2	1		1/2	1/4
1	0	1	1	1	0	1	.	0	1

64
16
8
4
1
12.5
6.25
93.25

5. B - THE BCD CODE FOR THE DECIMAL NUMBER 354 IS -- 0011 0101 0100.

In the BCD, or 8-4-2-1 code, each digit of a decimal number is represented by a group of four binary digits or bits called a code group. Depending upon the digit being coded, the four-bit code group will be one of the natural binary codes for the decimal numbers 0 to 9. Since 354 is a three digit number, there will be three groups of four binary digits. The binary equivalent of the decimal number 3 is 0011; for the decimal number 5, it is 0101; and for the decimal number 4, it is 0100. Therefore, 0011 0101 0100 is the BCD code for the decimal number 354.

6. D - THE XS-3 CODE FOR THE DECIMAL NUMBER 1,736 IS -- 0100 1010 0110 1001.

The XS-3 code is similar to the BCD (8-4-2-1) code in the respect that four bits are used to represent each digit in the decimal number being coded. The XS-3 code of a digit is the BCD code of the digit plus 3. The XS-3 code is formed from the BCD code by adding the BCD code for 3 to the BCD code for the decimal digit being coded. The addition is performed using the rules of binary addition.

1736 - decimal

```

0001 0111 0011 0110 - BCD
(+3) 0011 0011 0011 0011
-----
0100 1010 0110 1001 - XS-3

```

7. D - THE GRAY CODE FOR THE DECIMAL NUMBER 84 IS -- 1111110.

To find the Gray code for a given number, follow the following four steps:

1. Find the natural binary code for the number to be coded.
2. Write the number from Step 1 down twice, one over the other, as is done in arranging numbers which are to be added.
3. Shift the bottom number one place to the right and disregard the rightmost digit in the number.
4. Add the two numbers together using the following rules:

```

0 + 0 = 0
1 + 0 = 1
0 + 1 = 1
1 + 1 = 0

```

The sum produced by this process is the Gray code for the decimal number being coded.

```

Utilizing Step 1 - 1010100
Utilizing Step 2 - 1010100
                  1010100
-----
Utilizing Step 3 - 1010100
                  1010100
-----
Utilizing Step 4 - 1111110

```

8. C - ENCODING AND DECODING -- ARE INVERSE PROCESSES.

Encoding is the process of converting input decimal values into binary codes for entry into the computing system. Decoding is the process of converting binary information (used inside the computing system) to decimal information (for use outside the system).

9. A - GRAY CODING HAS THE CHARACTERISTIC OF -- CHANGING ONLY ONE BIT FROM ONE NUMBER TO THE NEXT.

In the Gray code the number 3 is 10 and 4 is 110. Only one bit changed in going from 3 to 4. In the natural binary code 3 is 11 and 4 is 100. Now three bits changed in going from 3 to 4.

10. B - A PARITY BIT IS ADDED TO A CODE TO -- MAKE THE CODE SELF-CHECKING.

A simple and effective method of error checking called PARITY CHECKING may be used to guard against an error in a code group.

In parity checking, an extra bit called a PARITY BIT is added to each code group. The parity bit is chosen to obtain either an even number of 1's in all code groups, or an odd number of 1's in all code groups.

QUESTIONS

IMPORTANT—These instructions **MUST** be accurately followed to avoid loss, or errors in grading.

Indicate your answer on this sheet by filling in the box for the most correct answer to each question, as illustrated in the example below.

When all questions have been answered, place the answer card in the proper position to line up the boxes on the card with the boxes on the sheet.

Next, copy the complete lesson code into the space provided on the card, and fill in the answer boxes to correspond with those previously filled in on this sheet.

Before mailing, be certain your correct student number, name and address appear on the card.

LESSON CODE
5230A

Example:

A	☐
B	☒
C	☐
D	☐

The natural satellite of the earth is the

(A) sun. (B) moon.

(C) planet Venus. (D) planet Mars.

- A ☐ B ☐ C ☐ D ☒

According to Equation 1 of this lesson, the actual value of the 3 in the decimal number 6731 is
(A) 0.3 (B) 3000 (C) 300 (D) 30
- A ☐ B ☐ C ☒ D ☐

The binary notation for the decimal number 73 is
(A) 11111. (B) 100101. (C) 1001001. (D) 1001101.
- A ☐ B ☐ C ☐ D ☒

The decimal notation for the binary number 1011001 is
(A) 98. (B) 73. (C) 37. (D) 89.
- A ☐ B ☐ C ☐ D ☒

The decimal notation for the binary number 1011101.01 is
(A) 55.5 (B) 25.25 (C) 57.125 (D) 93.25
- A ☐ B ☐ C ☐ D ☒

The BCD code for the decimal number 354 is
(A) 0110 1000 0111. (B) 0011 0101 0100. (C) 0110 0101 0111. (D) 0011 1000 0100.
- A ☐ B ☐ C ☐ D ☒

The XS-3 code for the decimal number 1736 is
(A) 0001 0111 0011 0110. (B) 0101 0011 0110 1000. (C) 0101 1100 1000 1011. (D) 0100 1010 0110 1001.
- A ☐ B ☐ C ☐ D ☒

The Gray code for the decimal number 84 is
(A) 1010100. (B) 1000 0100. (C) 1011 0111. (D) 1111110.
- A ☐ B ☐ C ☐ D ☒

Encoding and decoding
(A) are identical processes used to enter information into a computing system. (B) are identical processes used to extract information from a computing system. (C) are inverse processes. (D) are only used with the Gray code.
- A ☒ B ☐ C ☐ D ☐

Gray coding has the characteristic of
(A) changing only one bit in going from one number to the next. (B) a built-in parity bit. (C) built-in error checking. (D) representing each digit of the decimal number being coded with four binary digits.
- A ☐ B ☐ C ☒ D ☐

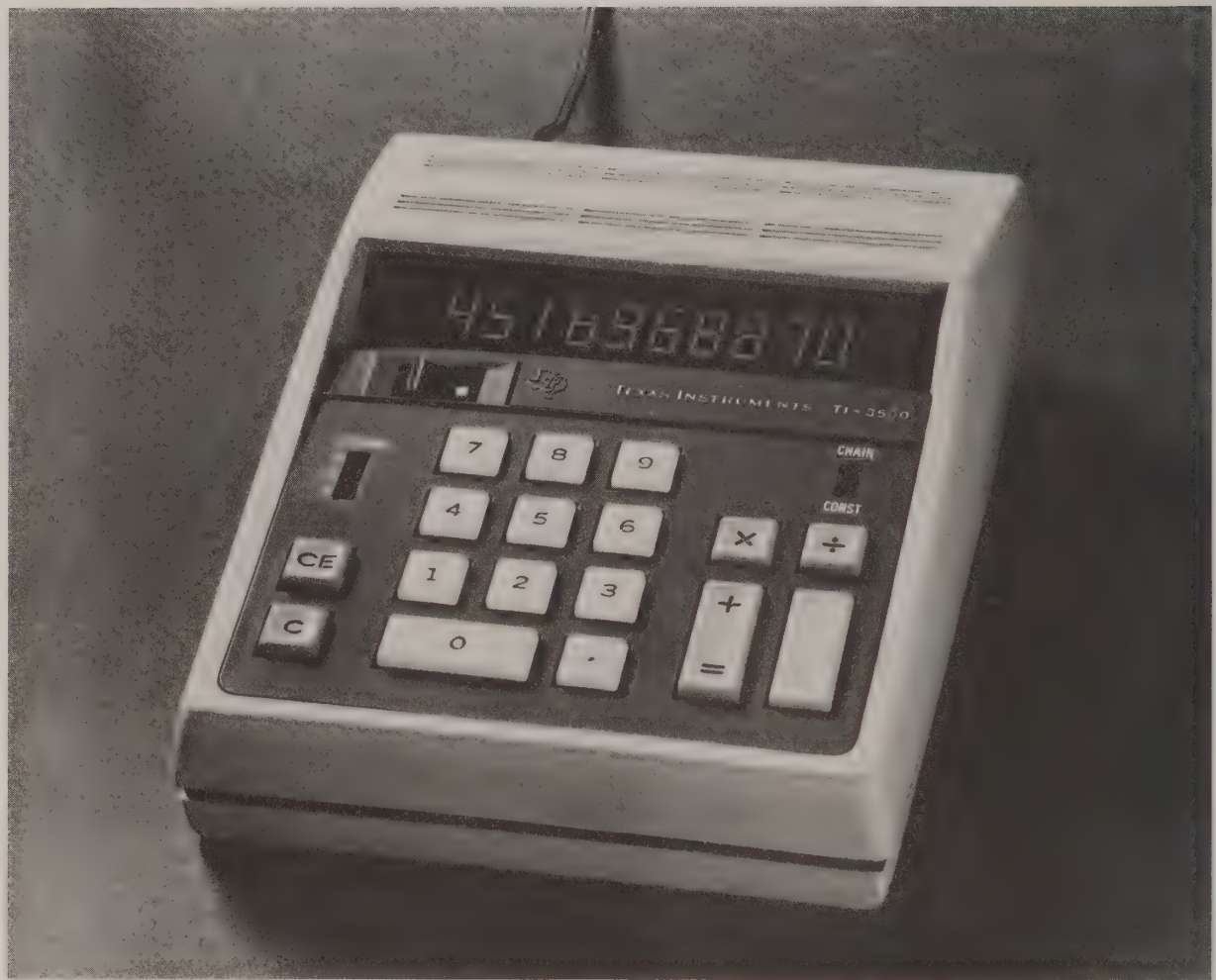
A parity bit is added to a code to
(A) decrease the number of bits required to code a given decimal number. (B) make the code self-checking. (C) make the code more efficient. (D) increase the chance of an error being introduced in the code.

BASIC ARITHMETIC CIRCUITS

5235

© COPYRIGHT 1973
BELL & HOWELL SCHOOLS, INC.
ALL RIGHTS RESERVED





For convenient performance of the basic arithmetic functions, this solid-state calculator is ideal. Its addition, subtraction, multiplication and division operations are virtually instantaneous. The calculator employs a single IC chip to perform the arithmetic computations; this chip is the equivalent of over 6000 transistors.

Courtesy Texas Instruments Inc.

CONTENTS

Basic Binary Arithmetic	Page 4
Binary Addition	Page 5
Binary Subtraction	Page 9
Negative and Complementary Numbers	Page 11
Complementary Subtraction	Page 13
Arithmetic Logic	Page 20
Adder Logic	Page 21
Subtractor Logic	Page 27
Complementary Subtractor Logic	Page 30
Binary Multiplication and Division	Page 32
Multiplication	Page 32
Division	Page 34

*Few things are impossible
to diligence and skill . . .
Great works are performed,
not by strength, but
perseverance.*

—Samuel Johnson

BASIC ARITHMETIC CIRCUITS

Digital systems are sometimes used to “control” signals and operations. The control system is a combination of on-off switches, channeling signals from one point to another, turning other equipment on and off, etc., according to a set of logical conditions. However, it is also possible to perform mathematical operations with logical devices. This is accomplished in sophisticated electronic computer systems which are, in fact, nothing more than combinations of on-off switches. For this role, it is necessary for a sequence of logical digits (0's and 1's) to be used as a code to represent specific numerical values.

One of the two main advantages in using binary numbers in electronic digital systems is the simplicity of the arithmetic operations. Whether you add, subtract, multiply, or divide, the process is always as simple as, or sometimes simpler than, the decimal counterpart. The other main advantage is that many electronic components and circuits have two stable states. These two stable states are represented by the two binary bits, 0 and 1.

While describing the arithmetic operations in this lesson, both the binary operation and the corresponding decimal operation will usually be shown. This comparison should make it clear how much alike the processes are.

In addition to acquainting you with the basic rules of binary arithmetic operations, we will describe a variety of logic combinations which allow these operations to be accomplished electronically.

Digital systems are replacing analog systems which you are most familiar with (radio, television, radar, etc.) at an ever increasing rate. This is especially true in space and military operations; however, this replacement is also beginning to have an impact on the industrial scene. With the digital approach, the analog functions of amplification, filtering, modulation, etc., are accomplished by performing the mathematical equivalent. For example, instead of continuously amplifying a signal with a transistor or vacuum tube, we may sample the signal periodically; convert the sample amplitude to a digital value; multiply this value by 10, 100, or 1000; and convert the new digital value into an amplified analog signal. Such procedures are revolutionizing the capability of electronic systems, and they will probably change the entire philosophy of electronic system design in the coming years.

BASIC BINARY ARITHMETIC

As in the decimal system, addition and subtraction are the basic operations performed in binary arithmetic. In decimal arithmetic, it is necessary for us to memorize all of the fundamental digit sums and digit differences in order

to perform addition and subtraction. In binary addition, it is necessary to memorize only four different combinations:

$$0 + 0 = 0 \quad (1a)$$

$$0 + 1 = 1 \quad (1b)$$

$$1 + 0 = 1 \quad (1c)$$

$$1 + 1 = 0 \text{ and a carry of } 1. \quad (1d)$$

There are also only four different subtraction combinations which must be memorized:

$$0 - 0 = 0 \quad (2a)$$

$$1 - 0 = 1 \quad (2b)$$

$$1 - 1 = 0 \quad (2c)$$

$$0 - 1 = 1 \text{ with a borrow of } 1. \quad (2d)$$

In this lesson, only Natural Binary Logic is used. That is, only weighted codes apply. Codes such as Gray and XS-3 must be converted to natural binary logic before the arithmetic operation can be accomplished.

We will describe the significance of the “carry” and “borrow” operations later in the lesson.

Binary Addition

Addition is very likely the first arithmetic operation performed by each of us in our early schooling. In its simplest form, it is basically a counting process. That is, we count up to the value of each number in succession and the sum is the total count. However, very early in our training we learn number combinations and no longer need to count each successive addition. By scanning down a column and recognizing the number combinations, the correct sum can be written.

For simple addition, where the sum for each column does not exceed the largest permissible symbol of the number system (the radix minus one, or 9 in the decimal system), the addition is nothing more than writing down the sum as found by recognizing the number combinations in each column. As examples:

<i>Binary</i>		<i>Decimal</i>
10		2
+101		+5
111	=	7

Note: In this lesson with examples such as the one just shown (where equivalent binary and decimal arithmetic is shown), the decimal numbers will be on the right of the binary numbers.

When binary addition involves the combination—0 + 1 or 1 + 0, as well as 0 + 0 (not shown in the example)—Equations 1a, 1b, and 1c apply. As long as the binary addition does not involve a carry, the respective bits of each binary number are simply added to obtain the bits for the sum. Further examples are as follows:

$$\begin{array}{rcl}
 \text{Binary} & & \text{Decimal} \\
 0001 & & 1 \\
 +0010 & = & +2 \\
 \hline
 0011 & & 3 \\
 \\
 0101 & & 5 \\
 +1000 & = & +8 \\
 \hline
 1101 & & 13
 \end{array}$$

Not much addition can be performed before we come across the sum of a column which exceeds the largest permissible digit (9 for the decimal system and 1 for the binary system). When the sum of a column exceeds the largest permissible digit, the right-hand digit is written down below the column and the remaining digits are “carried over” to the next columns. That is, the remaining digits are written in order above the columns to the left. A simple example showing the use of the **CARRY** is:

$$\begin{array}{rcl}
 \text{}^1 \text{ (Carry)} & & \text{}^1 \text{ (Carry)} \\
 10001 & & 17 \\
 + \quad 101 & = & + \quad 5 \\
 \hline
 10110 & & 22
 \end{array}$$

Other examples of increasing complexity include:

$$\begin{array}{rcl}
 \text{}^1 \text{ (Carry)} & & \\
 0010 & & 2 \\
 +0011 & = & +3 \\
 \hline
 0101 & & 5 \\
 \\
 \text{}^1 \text{ (Carry)} & & \\
 0101 & & 5 \\
 +0110 & = & + 6 \\
 \hline
 1011 & & 11 \\
 \\
 \text{}^{111} \text{ (Carry)} & & \\
 0011 & & 3 \\
 +0101 & = & +5 \\
 \hline
 1000 & & 8
 \end{array}$$

$$\begin{array}{r}
 \text{11 (Carry)} \\
 0110 \\
 + 0111 \\
 \hline
 1101
 \end{array}
 =
 \begin{array}{r}
 6 \\
 + 7 \\
 \hline
 13
 \end{array}$$

In the last example, there is a carry at the third bit. (The bits are numbered from right to left.) This carry plus the logical 1 in the third bit of the first number produces a 0 in the third bit, with 1 to carry. The carry is placed over the fourth bit. However, the third bit sum is not finished. The 0 (from the previous addition in this column) must be added to the 1 in the third bit of the second number, resulting in a 1 in the third bit of the answer ($1 + 1 + 1 = 0 + 1$, with carry = 1).

Although other methods can be devised to perform addition, the carry process is an important time saver. This is just as true for digital systems as it is for manual calculations. For this reason, considerable care is exercised to make this carry available and complete, enabling numbers to be added as quickly as possible. Although digital systems usually do not work with whole columns of figures (we add only one bit to another at a time), the carry is still needed. To illustrate by an extreme case:

$$\begin{array}{r}
 \text{1111111 (Carry)} \\
 1111111 \\
 + \quad 1 \\
 \hline
 10000000
 \end{array}
 =
 \begin{array}{r}
 127 \\
 + 1 \\
 \hline
 128
 \end{array}$$

Notice that each carry caused another. To save time, a computer must be able to complete this chain of carries quickly if high speed operation is to be realized.

As in decimal addition, it is possible to add three or more numbers:

$$\begin{array}{r}
 \text{11 (Carry)} \\
 1001 \\
 0100 \\
 + 0110 \\
 \hline
 10011
 \end{array}
 =
 \begin{array}{r}
 9 \\
 4 \\
 + 6 \\
 \hline
 19
 \end{array}$$

$$\begin{array}{r}
 \text{111 (Carry)} \\
 110 \\
 011 \\
 + 101 \\
 \hline
 1110
 \end{array}
 =
 \begin{array}{r}
 6 \\
 3 \\
 + 5 \\
 \hline
 14
 \end{array}$$

BASIC ARITHMETIC CIRCUITS

The addition of four 1's necessitates a carry of 10 (which is a carry two places to the left):

$$\begin{array}{r}
 \text{10 (Carry)} \\
 01 \\
 01 \\
 01 \\
 + 01 \\
 \hline
 100
 \end{array}
 =
 \begin{array}{r}
 1 \\
 1 \\
 1 \\
 + 1 \\
 \hline
 4
 \end{array}$$

In the addition of three binary numbers, you can be adding four or more 1's when the carries are considered.

$$\begin{array}{r}
 \text{1101 (Carry)} \\
 1010 \\
 0011 \\
 + 0111 \\
 \hline
 10100
 \end{array}
 =
 \begin{array}{r}
 10 \\
 3 \\
 + 7 \\
 \hline
 20
 \end{array}$$

$$\begin{array}{r}
 \text{1 (Carry)} \\
 \text{10101 (Carry)} \\
 1111 \\
 0111 \\
 + 1011 \\
 \hline
 100001
 \end{array}
 =
 \begin{array}{r}
 15 \\
 7 \\
 + 11 \\
 \hline
 33
 \end{array}$$

In the fourth bit (from the right) column of the last example we have two 1's carried over: the 10 from the second column and the 1 from the third. When six 1's are added, the result is 110, which means that a 1 is carried to the next column on the left and a 1 is also carried to the second column on the left.

$$\begin{array}{r}
 \text{11 (Carry)} \\
 01 \\
 01 \\
 01 \\
 01 \\
 01 \\
 01 \\
 + 01 \\
 \hline
 110
 \end{array}$$

Furthermore, the addition of eight 1's requires a carry of 1 to the third column on the left:

$$\begin{array}{r}
 100 \text{ (Carry)} \\
 01 \\
 01 \\
 01 \\
 01 \\
 01 \\
 01 \\
 01 \\
 01 \\
 \hline
 1000
 \end{array}$$

You would do well to work on your own binary examples and compare them with their equivalent decimal numbers.

Binary Subtraction

Subtraction of binary numbers follows rules similar to those associated with decimal subtraction. However, binary subtraction is far less involved than decimal subtraction. A simple example of subtraction in the binary number system is:

$$\begin{array}{r}
 1011001 \\
 - 10001 \\
 \hline
 1001000
 \end{array}
 =
 \begin{array}{r}
 89 \\
 - 17 \\
 \hline
 72
 \end{array}$$

As long as there is no borrowing (the inverse or complement of carrying), the binary subtraction process is straight-forward using Equations 2a, 2b, and 2c. The bit values are simply subtracted from one another as in ordinary arithmetic. Further examples are as follows:

$$\begin{array}{r}
 1011 \\
 - 1001 \\
 \hline
 0010
 \end{array}
 =
 \begin{array}{r}
 11 \\
 - 9 \\
 \hline
 2
 \end{array}$$

$$\begin{array}{r}
 10100 \\
 - 00100 \\
 \hline
 10000
 \end{array}
 =
 \begin{array}{r}
 20 \\
 - 4 \\
 \hline
 16
 \end{array}$$

The previous examples are common; however, it soon becomes necessary to **BORROW** just as in decimal arithmetic. That is, we go to the next digit

BASIC ARITHMETIC CIRCUITS

to the left, reduce this value by one and add the borrowed value to the column being solved. Examples are as follows:

$$\begin{array}{r} \text{}^0 \text{ (Borrow)} \\ 0\cancel{1}01 \\ -0010 \\ \hline 0011 \end{array} = \frac{5}{-2} = \frac{3}{3}$$

$$\begin{array}{r} \text{}^0 \text{ (Borrow)} \\ 0\cancel{1}011 \\ -00100 \\ \hline 00111 \end{array} = \frac{11}{-4} = \frac{7}{7}$$

$$\begin{array}{r} \text{}^0 \text{}^0 \text{ (Borrow)} \\ \cancel{1}0101\cancel{1}0 \\ -100101 \\ \hline 110001 \end{array} = \frac{7 \text{}^{16} \text{ (Borrow)}}{\cancel{8}\cancel{6}} = \frac{-37}{49}$$

For all of these examples, the bit to the left of the column being subtracted and needing a borrow is a 1. In some cases this may not be true. For example, one or more columns to the left may contain 0's before we reach a column containing a 1. In this case, we must go to the first column to the left containing a 1, reduce this 1 to a 0, and replace all intermediate 0's with 1's. For example:

$$\begin{array}{r} \text{}^0 \text{}^1 \text{ (Borrow)} \\ \cancel{1}\cancel{0}010 \\ -01110 \\ \hline 00100 \end{array} = \frac{18}{-14} = \frac{4}{4}$$

$$\begin{array}{r} \text{}^0 \text{}^1 \text{}^1 \text{ (Borrow)} \\ \cancel{1}\cancel{0}\cancel{0}01 \\ -01110 \\ \hline 00011 \end{array} = \frac{17}{-14} = \frac{3}{3}$$

Successive borrow operations may also be necessary:

$$\begin{array}{r} \text{}^0 \text{}^1 \text{}^1 \text{ (Second Borrow)} \\ \text{}^0 \text{ (First Borrow)} \\ \cancel{1}\cancel{0}\cancel{0}\cancel{1}0 \\ -01111 \\ \hline 00011 \end{array} = \frac{18}{-15} = \frac{3}{3}$$

$$\begin{array}{r} \text{}^0 \text{ (Second Borrow)} \\ \text{}^0 \text{}^1 \text{ (First Borrow)} \\ 110\cancel{1}\cancel{1}\cancel{0}0 \\ -100111 \\ \hline 1000101 \end{array} = \frac{9 \text{}^{18} \text{ (Borrow)}}{\cancel{1}\cancel{0}\cancel{8}} = \frac{-39}{69}$$

Negative and Complementary Numbers

The standard way of writing negative numbers is to place a “sign symbol” before a number if it is negative. For example, negative 41 is written as -41 . A sign symbol can also be placed before positive numbers such as $+41$. However, when no symbol appears before a number, the number is understood to be positive.

In digital systems, numbers are represented by a set of binary storage devices such as switches or flip-flops. Each storage device stores one bit. A set of storage devices which stores a number and is handled as one unit is called a **REGISTER**. A register made up of five switches can store any binary number from 00000 through 11111. For example, a closed switch may represent 1, and an open switch may represent 0. To also represent the binary numbers from 00000 through -11111 requires another bit and therefore another switch. This bit represents the sign ($-$ or $+$) of the number. For most digital systems utilizing this method of representing the sign of the number, if the sign bit is a 0, the number is positive; and if the sign bit is a 1, the number is negative.

Thus, a six-bit register is required to represent a five-place binary number and its sign. If the switch that represents the sign bit is open (0), the number is positive and has the magnitude of the number stored in the remaining five switches. If the switch which represents the sign bit is closed (1), the number is negative and again has the magnitude of the number stored in the remaining five switches. Figure 1 illustrates a six-bit register storing a five-place binary number and its sign. Register A contains the number -10 , and register B contains the number $+12$. In writing out the number stored in these registers, the sign bit is set apart from the magnitude of the number, usually with a radical point. Thus, the number stored in register A is 1.01010 , which represents -1010 binary or -10 decimal. The number stored in register B is 0.01100 , which represents $+1100$ binary or $+12$ decimal. To avoid confusion other symbols can be used to set the sign bit apart from the magnitude of the number, since the radical point is also used to indicate fractions. The hyphen ($-$) and star ($*$) are commonly used for this purpose. Thus, 1.01010 can also be written $1-01010$ or $1*01010$, and 0.01100 can also be written $0-01100$ or $0*01100$.

Many digital systems use another method to handle negative numbers in which a sign bit is not needed. In this method negative numbers are handled in **COMPLEMENTED FORM**. A complement is a quantity needed to complete or fill something. For example, when dealing with angles, a complement is the angle which must be added to another angle to form an angle of 90° . Thus, 30° is the complement of 60° , since $60^\circ + 30^\circ = 90^\circ$. Also, 60° is the complement of 30° , since $30^\circ + 60^\circ = 90^\circ$. By representing negative

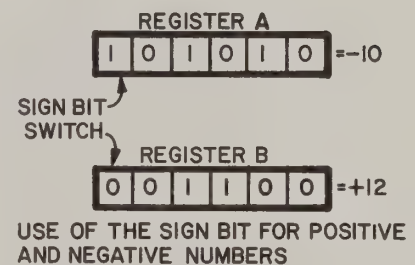


Figure
1

numbers in complemented form, a computer can use the same basic circuits for subtraction that it uses for addition.

There are two main types of complements which are useful in the binary number system, as well as in the decimal system. These complements are called the **TRUE** (or real) **COMPLEMENT** and the **RADIX-MINUS-ONE COMPLEMENT**. The true complement is called the 10's complement in the decimal system and the 2's complement in the binary system. The radix-minus-one complement is called the 9's ($10 - 1 = 9$) complement in the decimal system and the 1's ($2 - 1 = 1$) complement in the binary system.

In the decimal system, the 10's complement of any number may be formed by subtracting each digit of the number from 9 and then adding 1 to the far right digit of the number formed. The far right digit is called the least significant digit, since its value is less than the value of any other digits in the number. In the decimal system, the 9's complement of a number is formed by subtracting each digit in the number from 9. For example, the 9's complement of 67 is 32, and the 9's complement of 39 is 60. Also, the 10's complement of 67 is 33, and the 10's complement of 39 is 61. These complements are formed as follows:

$\begin{array}{r} 99 \\ -67 \\ \hline 32 \\ + 1 \\ \hline 33 \end{array}$	(9's complement) (10's complement)	$\begin{array}{r} 99 \\ -39 \\ \hline 60 \\ + 1 \\ \hline 61 \end{array}$
---	---	---

In the binary system, the 1's complement of a number is formed by subtracting each bit of the number from 1. For example, the 1's complement of 11011 is 00100, and the 1's complement of 111000 is 000111. Furthermore, in the binary system, the 2's complement of any number may be formed by subtracting each bit of the number from 1 and then adding 1 to the far right bit (least significant bit) of the number formed, or by adding 1 to the 1's complement. For example, the 2's complement of 11011 is 00101, and the 2's complement of 111000 is 001000. These complements (1's and 2's) are formed as follows:

$\begin{array}{r} 11111 \\ -11011 \\ \hline 00100 \\ + 1 \\ \hline 00101 \end{array}$	(1's complement) (2's complement)	$\begin{array}{r} 111111 \\ -111000 \\ \hline 000111 \\ + 1 \\ \hline 001000 \end{array}$
---	--	---

Notice that in the decimal system, the 10's complement is actually formed by adding 1 to the 9's complement. In the binary system, the 2's comple-

ment is actually formed by adding 1 to the 1's complement. Also notice that although complements may be formed by subtraction in paper calculations, in the binary system the 1's complement may be formed by simply replacing the 1's with 0's and the 0's with 1's. Thus, in electronic circuits a number can be complemented by simply changing the state of each two-stable-state device. For example, to complement a number stored in a flip-flop register, each flip-flop is switched to its other state. This produces the 1's complement of the original number.

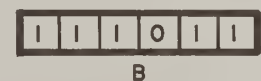
When numbers are written in a normal manner, 0's to the left of a whole number or to the right of a decimal or binary fraction are not significant and can be dropped without changing the value of the number. However, when dealing with complemented numbers, these 0's must be retained. For example, the 1's complement of 111000 is 000111, and the 1's complement of 1000 is 0111. If it were not for the different number of 0's here, there would be no way to tell the two complements apart.

In digital systems, a number stored in a register may be complemented by simply changing the stable state of each two-stable-state device in the register. The number stored in a register is represented by the complete contents of the register, including any 0's to the left of a whole number or to the right of a decimal or binary fraction. Figure 2A represents a six-bit register which contains the binary number 000100. The actual value of this is 100 binary or 4 decimal. However, when dealing with complementary numbers, the 0's to the left of the 1 must also be considered since these 0's become 1's when the 1's complement of the number is formed, as shown in Figure 2B (111011 or 59 in decimal). Figure 3 shows this same idea with a seven-bit register. Here, the contents of the register are 0000100. When the 1's complement of this number is formed, the result is 1111011 (123 in decimal).

If the number of two-stable-state devices in a register is not specified, you can usually disregard any 0's to the left of a whole number or to the right of a decimal or binary fraction. This is usually done in paper calculations. However, keep in mind that these 0's are present whenever the number of two-stable-state devices in a register is greater than the number of significant places in the binary number.

Complementary Subtraction

From now on we will not include "(Carry)" and "(Borrow)" in the examples or Practice Exercises. If you have difficulty because of this in the rest



DEVELOPMENT OF 1'S
COMPLEMENT FOR SIX
BIT BINARY NUMBER

Figure
2



DEVELOPMENT OF 1'S COM-
PLEMENT FOR SEVEN BIT
NUMBER

Figure
3

of the lesson, you should review the material that covers borrowing and carrying.

The use of complemented numbers allows a computer to perform subtraction with addition circuitry. This allows the system to be implemented with a minimum number of logical elements. Either the true complement or the radix-minus-one complement may be used for subtraction. The four steps utilizing the true complement to subtract one number (lower or subtrahend) from another (upper or minuend) are:

1. Obtain the true complement of the subtrahend. Note: The complement must have the same number of places as the minuend.
2. Add the complement to the minuend.
3. If the addition of the left column develops a carry, discard it and replace it with a plus sign. You have obtained the desired answer (the minuend is larger than the subtrahend).
4. If the addition of the left column does not develop a carry, obtain the true complement of the result. The complement is the desired answer and it is negative (the subtrahend was larger than the minuend).

The following examples show how these steps are used to subtract with the true complement. (Remember, in the decimal system the true complement is called the 10's complement, and in the binary system the true complement is called the 2's complement.) Below is an example of using the 10's complement to subtract decimal numbers:

Ordinary Decimal
Subtraction

$$\begin{array}{r} 87 \text{ (Minuend)} \\ -56 \text{ (Subtrahend)} \\ \hline +31 \text{ (Difference)} \end{array}$$

Addition of
10's Complement

$$\begin{array}{r} 87 \\ + 44 \\ \hline +131 \end{array}$$



Carry is Dropped.

According to Step 1, in this example you must first obtain the 10's complement of 56. Note: The complement must have the same number of places as the minuend. The 10's complement of 56 is 44 ($99 - 56 = 43$, and $43 + 1 = 44$). Then, according to Step 2, add 44 to 87 to obtain 131. Finally, by Step 3, the carry is dropped and replaced with a plus sign to obtain an answer of +31.

Another example of using the 10's complement to subtract decimal numbers is:

Ordinary Decimal Subtraction	Addition of 10's Complement
---------------------------------	--------------------------------

$$\begin{array}{r} 56 \\ -95 \\ \hline -39 \end{array}$$

$$\begin{array}{r} 56 \\ +05 \\ \hline 61 \end{array}$$

$$\begin{array}{r} \text{Obtaining 10's} \quad 99 \\ \text{Complement of} \quad -61 \\ 61 \quad \quad \quad \hline 38 \\ + 1 \\ \hline -39 \end{array}$$

According to Step 1, in this example you must first obtain the 10's complement of 95 and the complement must have the same number of places as the minuend. The 10's complement of 95 is 05 ($99 - 95 = 04$, and $04 + 1 = 05$). Then, according to Step 2, add 05 to 56 to obtain 61. Finally, by Step 4, since no carry is developed by the addition of the left column, obtain the 10's complement of 61, which is 39, and -39 is the correct answer.

Below is an example of using the 2's complement to subtract binary numbers:

Ordinary Binary Subtraction	Addition of 2's Complement
--------------------------------	-------------------------------

$$\begin{array}{r} 111001 \\ -011101 \\ \hline 011100 \end{array}$$

$$\begin{array}{r} 111001 \\ + 100011 \\ \hline +1011100 \end{array}$$

↓
Carry is Dropped.

According to Step 1, in this example you must first obtain the 2's complement of 011101, remembering that the complement must have the same number of places as the minuend. The 2's complement of 011101 is 100011 (the 0's are changed to 1's, the 1's are changed to 0's, and 1 is added to the least significant bit). Then, according to Step 2, add 100011 to 111001 to obtain 1011100. Finally, by Step 3, the carry is dropped and replaced with a plus sign to obtain an answer of +011100.

Another example of using the 2's complement to subtract binary numbers is (where the subtrahend is larger than the minuend):

Ordinary Binary Subtraction	Addition of 2's Complement
$\begin{array}{r} 100101 \\ -110100 \\ \hline -001111 \end{array}$	$\begin{array}{r} 100101 \\ +001100 \\ +110001 \\ \hline \end{array}$
Obtaining 2's Complement of 110001	$\begin{array}{r} 001110 \\ + \quad 1 \\ \hline -001111 \end{array}$

According to Step 1, in this example you must first obtain the 2's complement of 110100 and the complement must have the same number of places as the minuend. The 2's complement of 110100 is 001100 (the 0's are changed to 1's, the 1's are changed to 0's, and 1 is added to the least significant bit). Then, according to Step 2, add 001100 to 100101 to obtain 110001. Finally, by Step 4, since no carry is developed by the addition of the left column, obtain the 2's complement of 110001, which is 001111, and -001111 is the correct answer.

A few more examples of using the true complement to subtract are:

Ordinary Decimal Subtraction	Addition of 10's Complement
$\begin{array}{r} 4637 \\ -0805 \\ \hline +3832 \end{array}$	$\begin{array}{r} 4637 \\ +9195 \\ \hline +13832 \end{array}$ ↓ Carry is Dropped.

Ordinary Decimal Subtraction	Addition of 10's Complement
$\begin{array}{r} 23 \\ -56 \\ \hline -33 \end{array}$	$\begin{array}{r} 23 \\ +44 \\ \hline 67 \end{array}$
Obtaining 10's Complement of 67	$\begin{array}{r} 99 \\ -67 \\ \hline 32 \\ + 1 \\ \hline -33 \end{array}$

Ordinary Binary Subtraction

$$\begin{array}{r} 101110 \\ -100110 \\ \hline +001000 \end{array}$$

Addition of 2's Complement

$$\begin{array}{r} 101110 \\ + 011010 \\ \hline 1001000 \end{array}$$



Carry is Dropped.

Ordinary Binary Subtraction

$$\begin{array}{r} 010000 \\ -100000 \\ \hline -010000 \end{array}$$

Addition of 2's Complement

$$\begin{array}{r} 010000 \\ +100000 \\ \hline 110000 \end{array}$$

Obtaining 2's Complement of 110000

$$\begin{array}{r} +001111 \\ + \quad 1 \\ \hline -010000 \end{array}$$

The steps for using the radix-minus-one complement to subtract are:

1. Obtain the radix-minus-one complement of the subtrahend. Note: The complement must have the same number of places as the minuend.
2. Add the complement to the minuend.
3. If the addition of the left column develops a carry, add it to the far right (least significant) digit or bit in the result. This is called an end-around carry. The addition of the end-around carry gives the desired answer, and it is positive.
4. If the addition of the left column does not develop a carry, obtain the radix-minus-one complement of the result. The complement is the desired answer, and it is negative.

The following examples show how these steps are used to subtract using the radix-minus-one complement. An example of using the 9's complement to subtract decimal numbers is:

Ordinary Decimal Subtraction

$$\begin{array}{r} 87 \\ -56 \\ \hline +31 \end{array}$$

Addition of 9's Complement

$$\begin{array}{r} 87 \\ +43 \\ \hline 130 \\ \downarrow 1 \\ +31 \end{array}$$

According to Step 1, in this example you must first obtain the 9's complement of 56 and the complement must have the same number of places as the minuend. The 9's complement of 56 is 43 ($99 - 56 = 43$). Then, according to Step 2, add 43 to 87 to obtain 130. Finally, by Step 3, the end-around carry is added to the least significant digit to obtain an answer of +31.

Another example of using the 9's complement to subtract decimal numbers is:

Ordinary Decimal Subtraction	Addition of 9's Complement
$\begin{array}{r} 56 \\ -95 \\ \hline -39 \end{array}$	$\begin{array}{r} 56 \\ +04 \\ \hline 60 \end{array}$
Obtaining 9's Complement of 60	$\begin{array}{r} 99 \\ -60 \\ \hline -39 \end{array}$

According to Step 1, in this example you must first obtain the 9's complement of 95 and the complement must have the same number of places as the minuend. The 9's complement of 95 is 04 ($99 - 95 = 04$). Then, according to Step 2, add 04 to 56 to obtain 60. Finally, by Step 4, since no end-around carry is developed, obtain the 9's complement of 60, which is 39, and -39 is the correct answer.

An example of using the 1's complement to subtract binary numbers is:

Ordinary Binary Subtraction	Addition of 1's Complement
$\begin{array}{r} 111001 \\ -011101 \\ \hline +011100 \end{array}$	$\begin{array}{r} 111001 \\ + 100010 \\ \hline 1011011 \\ \text{ } \rightarrow 1 \\ \hline + 011100 \end{array}$

According to Step 1, in this example you must first obtain the 1's complement of 011101 and the complement must have the same number of places as the minuend. The 1's complement of 011101 is 100010 (the 0's are changed to 1's and the 1's are changed to 0's). Then, according to Step 2, add 100010 to 111001 to obtain 1011011. Finally, by Step 3, the end-

around carry is added to the least significant bit to obtain an answer of +011100.

Another example of using the 1's complement to subtract binary numbers is:

Ordinary Binary Subtraction	Addition of 1's Complement
$\begin{array}{r} 100101 \\ -110100 \\ \hline -001111 \end{array}$	$\begin{array}{r} 100101 \\ +001011 \\ \hline 110000 \end{array}$
Obtaining 1's Complement of 110000	-001111

According to Step 1, in this example you must first obtain the 1's complement of 110100 and the complement must have the same number of places as the minuend. The 1's complement of 110100 is 001011 (the 0's are changed to 1's and the 1's are changed to 0's). Then, according to Step 2, add 001011 to 100101 to obtain 110000. Finally, by Step 4, since no end-around carry is developed, obtain the 1's complement of 110000, which is 001111, and -001111 is the correct answer.

A few more examples of using the radix-minus-one complement to subtract are:

Ordinary Decimal Subtraction	Addition of 9's Complement
$\begin{array}{r} 4637 \\ -0805 \\ \hline +3832 \end{array}$	$\begin{array}{r} 4637 \\ + 9194 \\ \hline 13831 \\ \text{└─→ 1} \\ + 3832 \end{array}$
Ordinary Decimal Subtraction	Addition of 9's Complement
$\begin{array}{r} 23 \\ -56 \\ \hline -33 \end{array}$	$\begin{array}{r} 23 \\ +43 \\ \hline 66 \end{array}$
Obtaining 9's Complement of 66	$\begin{array}{r} 99 \\ -66 \\ \hline -33 \end{array}$

BASIC ARITHMETIC CIRCUITS

Ordinary Binary Subtraction

$$\begin{array}{r} 101110 \\ -100110 \\ +001000 \\ \hline \end{array}$$

Addition of 1's Complement

$$\begin{array}{r} 101110 \\ + 011001 \\ \hline 1000111 \\ \xrightarrow{\quad} 1 \\ + 001000 \\ \hline \end{array}$$

Ordinary Binary Subtraction

$$\begin{array}{r} 010000 \\ -100000 \\ -010000 \\ \hline \end{array}$$

Addition of 1's Complement

$$\begin{array}{r} 010000 \\ +011111 \\ \hline 101111 \\ \hline \end{array}$$

Obtaining 1's Complement
of 101111

$$-010000$$

ARITHMETIC LOGIC

Inputs		Outputs	
A	B	S	C ₀
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$\begin{aligned} A \cdot \bar{B} + \bar{A} \cdot B &= S \\ A \cdot B &= C_0 \end{aligned}$$

Truth Data for Half
Adder

Table
1

The manner in which logic circuits can be combined to perform binary addition and subtraction can be best understood by putting Equations 1a through 1d and 2a through 2d into truth tables.

For binary addition, Table 1 is used. The two inputs are A and B; the output S is the "sum" output, and the output C₀ is for "carry." Notice that the first row came from Equation 1a, the second row from 1b, the third row from 1c, and the last row from 1d. Also notice that the inputs A and B and the output S behave as an EXCLUSIVE OR function, and notice that the output C₀ performs the AND function. Hence, the basic binary adder logic is a two-input, two-output device: one output representing the binary sum (EXCLUSIVE-OR function), and the other representing the carry (AND function). The logic formula for the sum is:

$$A \cdot \bar{B} + \bar{A} \cdot B = S \quad (\text{EXCLUSIVE-OR})$$

and the formula for the carry is:

$$A \cdot B = C_0 \quad (\text{AND}).$$

A very similar development for binary subtraction can be made. Table 2 is used for binary subtraction. The first row came from Equation 2a, the second row from Equation 2d, the third row from Equation 2b, and the last row from Equation 2c. The two inputs are, again, A and B, where A

The following Practice Exercise questions cover the subjects which you have just studied. They are:

BASIC BINARY ARITHMETIC

BINARY ADDITION

BINARY SUBTRACTION

NEGATIVE AND COMPLEMENTARY NUMBERS

COMPLEMENTARY SUBTRACTION

1. Digital computer systems usually work with information that is in binary form. True or False?

2. Find the following binary sums:

$$\begin{array}{r} \text{(a)} \quad 00010 \\ +00100 \\ \hline \end{array} \quad \begin{array}{r} \text{(b)} \quad 00011 \\ +01000 \\ \hline \end{array} \quad \begin{array}{r} \text{(c)} \quad 00110 \\ +01001 \\ \hline \end{array}$$

$$\begin{array}{r} \text{(d)} \quad 00011 \\ +00110 \\ \hline \end{array} \quad \begin{array}{r} \text{(e)} \quad 01010 \\ +00011 \\ \hline \end{array} \quad \begin{array}{r} \text{(f)} \quad 00111 \\ +00011 \\ \hline \end{array}$$

3. Convert the binary additions in Practice Exercise 2 into decimal additions and find the decimal sums.

4. Find the following binary differences:

$$\begin{array}{r} \text{(a)} \quad 01111 \\ -01100 \\ \hline \end{array} \quad \begin{array}{r} \text{(b)} \quad 01001 \\ -01000 \\ \hline \end{array} \quad \begin{array}{r} \text{(c)} \quad 10011 \\ -00011 \\ \hline \end{array}$$

$$\begin{array}{r} \text{(d)} \quad 01100 \\ -01001 \\ \hline \end{array} \quad \begin{array}{r} \text{(e)} \quad 10000 \\ -00010 \\ \hline \end{array} \quad \begin{array}{r} \text{(f)} \quad 01001 \\ -00111 \\ \hline \end{array}$$

5. Convert the binary subtractions in Practice Exercise 4 into decimal subtractions and find the decimal differences.

6. In some computers, an extra bit is placed at the right-hand end of a binary code group in order to indicate the sign of the number. True or False?

7. A second method for representing negative numbers in computers is to convert the number to its _____ form.

8. There are two types of complemented forms. Name them.

9. Write the true 2's complement for the subtrahend binary numbers in Practice Exercise 4.

10. Write the 1's complement for the subtrahend binary numbers in Practice Exercise 4.
11. Perform the binary subtractions for (a), (b) and (c) of Practice Exercise 4 using the true complement of the subtrahends.
12. Perform the binary subtractions for (d), (e) and (f) of Practice Exercise 4 using the 1's complement of the subtrahends.
13. Find the following binary differences:

(a)	10111	(b)	01010
	<u>-11000</u>		<u>-10101</u>

is the minuend and B is the subtrahend. The output D represents the "difference." When the subtrahend B is larger than the minuend A, we have: $A = 0$ and $B = 1$; and "borrowing" is necessary. Note that the output D behaves in the same manner, in relation to inputs A and B, as did S in the adder. That is, D also behaves as an EXCLUSIVE-OR function. However, the "borrow" output behaves as an INHIBIT function, where A is the inhibit input and B is the signal input. As with the binary adder, the binary subtractor is a two-input, two-output device. One output represents the binary difference (EXCLUSIVE-OR function), but the other represents the "borrow" (INHIBIT function). In the subtractor a distinction must be made between the inputs: one is the minuend (input A) while the other is the subtrahend (input B). In other words, B is taken away from A. When B is larger than A, borrowing is necessary in a number system of any radix.

Inputs		Outputs	
A	B	D	B ₀
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$$A \cdot \bar{B} + \bar{A} \cdot B = D$$

$$\bar{A} \cdot B = B_0$$

Truth Data for Half Subtractor

Table
2

The logic formula for the difference is:

$$A\bar{B} + \bar{A}B = D,$$

and the formula for the borrow is:

$$\bar{A} \cdot B = B_0.$$

Subtraction by complemented addition is easily accomplished with logical elements. This is especially true if the 1's complement is used. In this case, the complement is obtained by inverting the subtrahend (passing it through a NOT circuit). The addition is then performed by a device with the truth data shown in Table 1. Finally, the result must be applied to a logic arrangement which implements the steps for the left-column "carry" or "no carry" condition. Of course, for the 1's complement, as long as the subtrahend is not larger than the minuend, only the "carry" condition applies. (Provision must be made to delete the left-most bit and add it to the right-most bit.)

Adder Logic

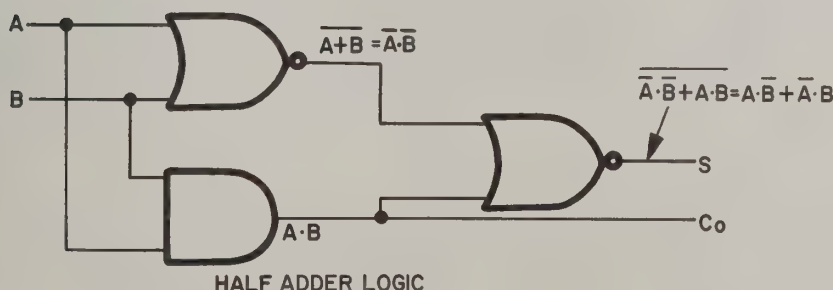


Figure 4

BASIC ARITHMETIC CIRCUITS

Figure 4 illustrates the logic arrangement known as the **HALF ADDER**. The EXCLUSIVE-OR function ($A \cdot \bar{B} + \bar{A} \cdot B$) is available at the sum (S) output terminal. A second output is available, from the first-level AND function, to provide the carry condition. Since this is simply a direct output from an AND gate, $C_o = A \cdot B$. Again, the truth data is presented in Table 1.

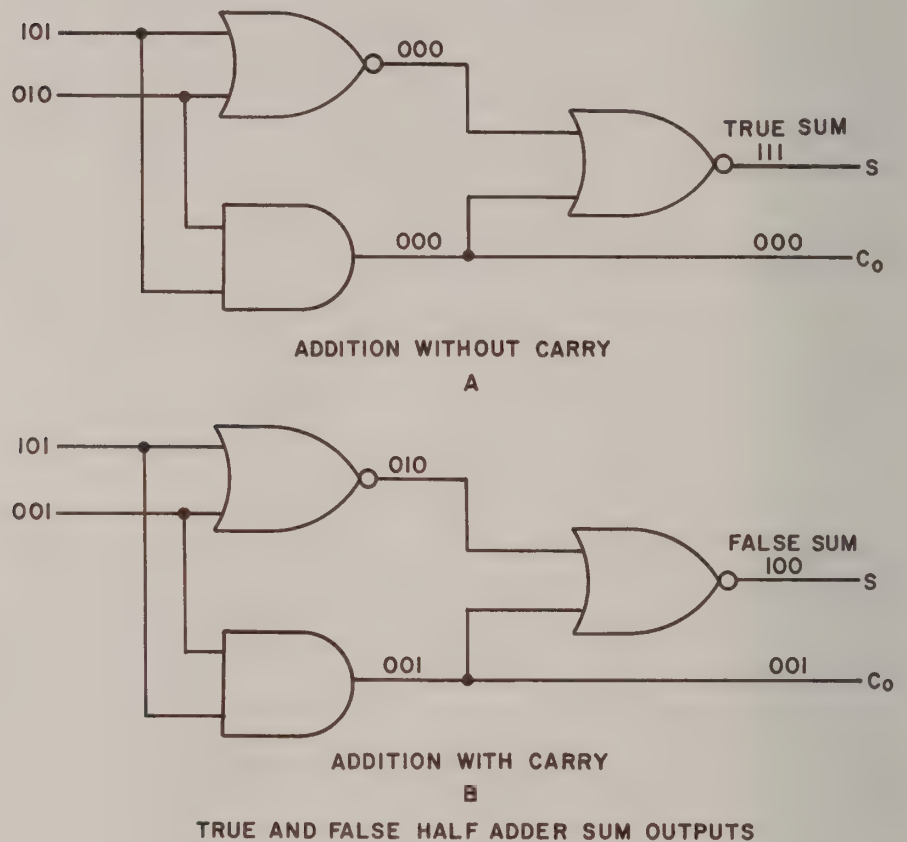
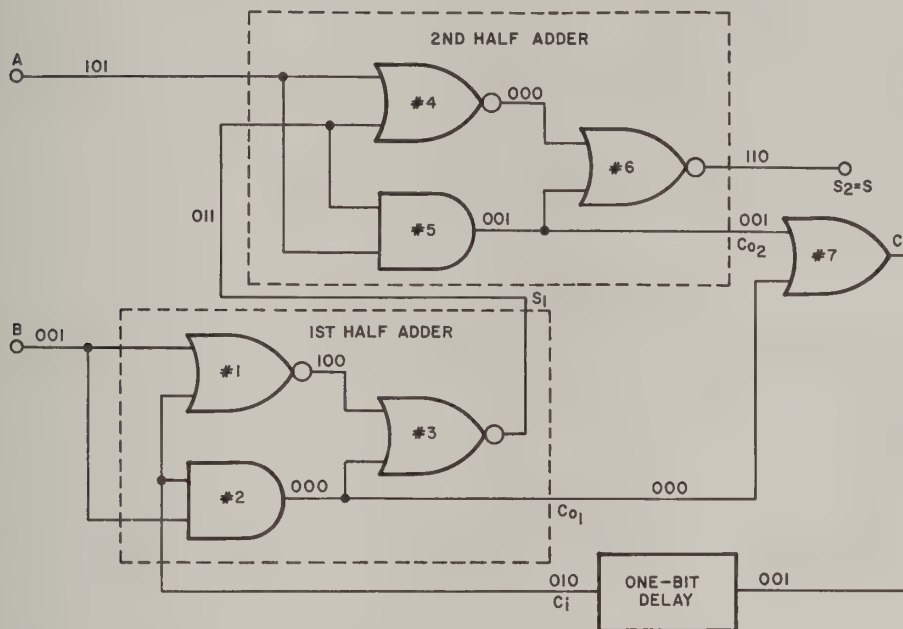


Figure 5

A single HALF ADDER, by itself, is not sufficient to perform all possible binary additions. The sum output will only be a "TRUE SUM" as long as no carry is involved (Figure 5A). When there is a carry, the sum output is a "FALSE SUM." This is shown in Figure 5B. The correct sum in Figure 5B would be:

$$\begin{array}{r} 1 \\ 101 \\ +001 \\ \hline 110 \end{array}$$

In order to perform the addition of all possible binary numbers, the C_0 information must be shifted one position to the left and must be added to the S information. This can be accomplished with an adder which has provisions for three inputs; two to handle the binary numbers being added, the third to add in a delayed version of the C_0 information. This is known as a **FULL ADDER**.



SERIAL HALF ADDER ARRANGEMENT TO OBTAIN THE FULL ADDER FUNCTION

Figure 6

Two half adders may be arranged serially, as shown in Figure 6, to form a **FULL ADDER**. Serial means that the operations are performed in sequence. That is, one half adder performs its operations; its outputs become inputs for the second half adder, which then performs its own operations. The circuit of Figure 4 is used for the two half adders. Other half adder circuits may be used instead of the circuit of Figure 4. The binary addition which produced a false sum for the half adder is shown to provide a true sum with the full adder. However, the additional C_1 information must be provided if the full adder is to provide the true sum. This information may be supplied by feeding C_0 back to C_1 through a single timing interval delay device (a one-bit time delay).

The truth table for the full adder can be constructed, level by level, using Table 1 and Figure 6. The development is shown in Table 3, with the three input columns and the two output columns selected to form the final truth

BASIC ARITHMETIC CIRCUITS

A	B	C _i	S	C _o
1	1	1	1	1
1	1	0	0	1
1	0	1	0	1
1	0	0	1	0
0	1	0	1	0
0	0	1	1	0
0	1	1	0	1
0	0	0	0	0

$$\begin{aligned} & \bar{A} \cdot \bar{B} \cdot \bar{C}_i + \bar{A} \cdot B \cdot \bar{C}_i + \\ & \bar{A} \cdot \bar{B} \cdot C_i + A \cdot \bar{B} \cdot C_i = S \\ & A \cdot B + A \cdot C_i + B \cdot C_i = C_o \end{aligned}$$

Final Truth Table for Full Adder

Table
4

Inputs			First Half Adder Outputs		Second Half Adder Outputs		Final Carry Out
A*	B [#]	C _i [#]	S ₁ *	C _{o1} **	S ^{##}	C _{o2} **	C _o
1	1	1	0	1	1	0	1
1	1	0	1	0	0	1	1
1	0	1	1	0	0	1	1
1	0	0	0	0	1	0	0
0	1	0	1	0	1	0	0
0	0	1	1	0	1	0	0
0	1	1	0	1	0	0	1
0	0	0	0	0	0	0	0

* Second Half Adder Inputs
First Half Adder Inputs
Output of the Full Adder

** Inputs for Final OR gate
(Carry-Out)

Construction of Truth Data for Full Adder

Table 3

data in Table 4. Using those combinations which provide a sum output of 1, the full adder sum-of-products expression is:

$$S = \bar{A} \cdot \bar{B} \cdot C_i + \bar{A} \cdot B \cdot \bar{C}_i + A \cdot \bar{B} \cdot \bar{C}_i + A \cdot B \cdot C_i \quad (3)$$

Using those combinations which provide a carry output of 1, the full adder carry output sum-of-products expression is:

$$C_o = \bar{A} \cdot B \cdot C_i + A \cdot \bar{B} \cdot \bar{C}_i + A \cdot \bar{B} \cdot C_i + A \cdot B \cdot C_i$$

which reduces to

$$C_o = A \cdot B + A \cdot C_i + B \cdot C_i \quad (4)$$

because

$$\begin{aligned} \bar{A} \cdot B \cdot C_i + A \cdot B \cdot C_i &= (\bar{A} + A)B \cdot C_i = BC_i \\ A \cdot \bar{B} \cdot \bar{C}_i + A \cdot B \cdot \bar{C}_i &= AB(\bar{C}_i + C_i) = AB, \text{ and} \\ A \cdot \bar{B} \cdot C_i + A \cdot B \cdot C_i &= AC_i(\bar{B} + B) = AC_i. \end{aligned}$$

Figure 6 shows the full adder with an input of 101 (decimal 5) and 001 (decimal 1) at terminals A and B, respectively. In order to analyze the circuit with these inputs, start with the lowest weight bit (on the right) at A and at B (where each input has a 1). Since the one-bit delay adds a 0 to the right of the number representing the output of gate 7, we will assume that the rightmost digit out of the delay is a 0. Gates 1 and 2 have inputs of 1 and 0 and the

output of each is 0. Gate 3 is a NOR and its right-most output is 1. Gates 4 and 5 have a 1 at each input. The output of gate 4 is a 0 at the right-most digit, while gate 5 has a 1. Then gate 6 (a NOR) has inputs of 0 and 1 and consequently its first output digit (on the right) is a 0. Gate 7 (an OR) receives a 1 from gate 5 and a 0 from gate 2; thus, the output of gate 7 is a 1. The delay circuit delays the 1 until the next bit is applied to the A and B inputs.

Now we can proceed evaluating the middle digit in the circuit. Gates 1 and 2 have inputs from terminal B of 0 and at C_1 of 1. The output of both gates is another 0; consequently, gate 3 has another 1 output. Gates 4 and 5 then have 1 and 0 inputs; their outputs are thus both zeros. Gate 6 then has a 1 for the second digit. Gate 7 has a 0 output this time, since the inputs are both 0's.

With a 0 at the input of the delay circuit, its output is also 0 when the third digit in our example is applied to inputs A and B. Now we can quickly determine the logic levels in the circuit of Figure 6 with the third digit from the right (the last in this case). Gates 1 and 2 both have 0's at their inputs; gate 1 then has a 1 output, while gate 2 has a 0. Gate 3 has an output of 0. Gates 4 and 5 have 0 outputs since the inputs are 1 and 0. The final output is a 1 at S, making the complete sum of 110, which is a decimal 6 when the inputs are 101 and 001 for decimals 5 and 1, respectively.

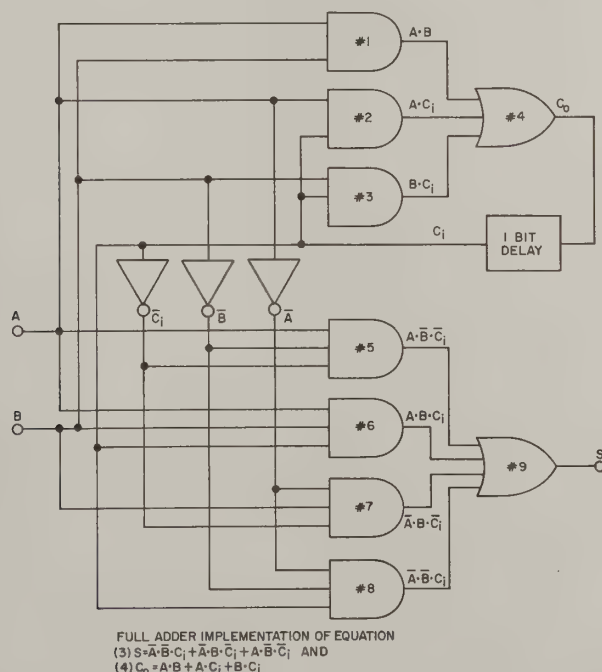
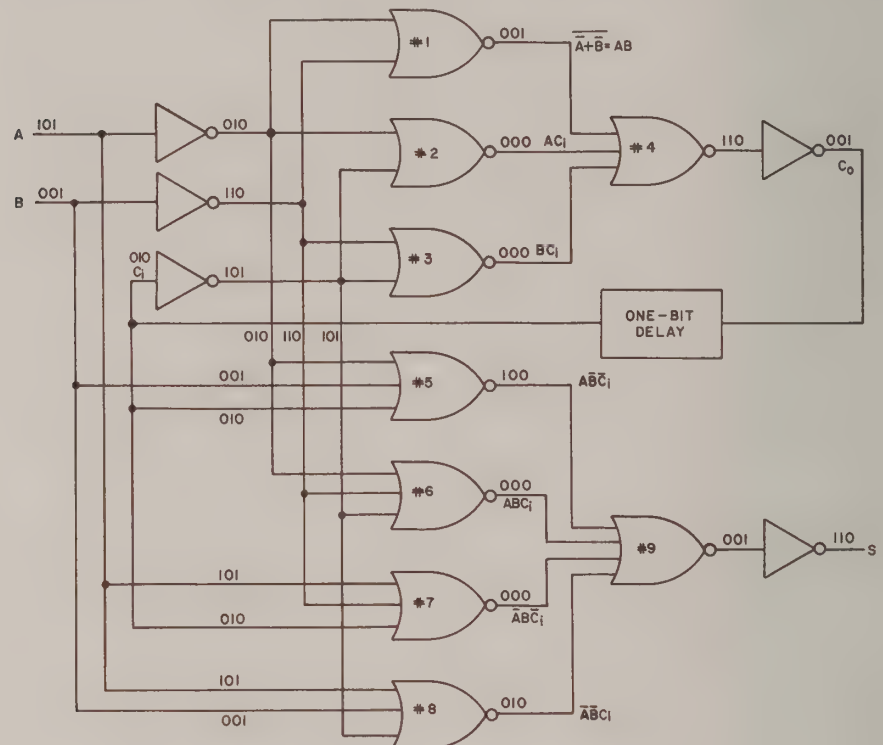


Figure 7

Figure 7 shows a direct implementation of Equations 3 and 4, using AND, OR, and NOT circuits and the 1-bit delay network. Gates 1 through 4 implement Equation 4, while the rest of the gates implement Equation 3. Note that the inputs to AND gate 1 are A and B , while the inputs to AND gate 5 are A , B , and $\bar{C}_1$.

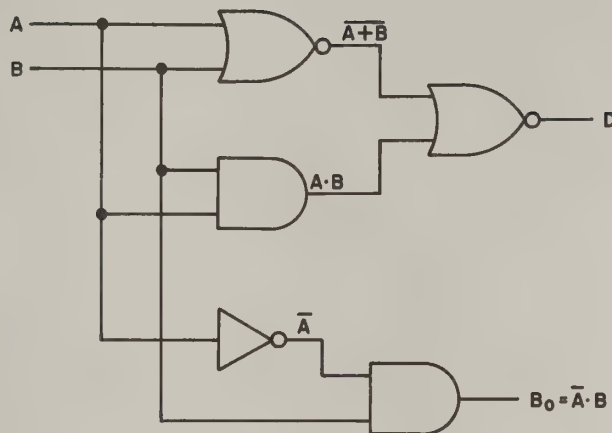


AN ALTERNATE LOGIC ARRANGEMENT FOR THE FULL ADDER

Figure 8

Figure 8 is very similar. All of the circuits except the delay have inversion at the output, which means that CE amplifiers can be used in each NOR gate and inverter. Notice that gate 1 (a NOR) has inputs of $\bar{A}$ and $\bar{B}$, but since it is a NOR gate, its output is $\bar{\bar{A} + \bar{B}}$, which is the same as $A \cdot B$. Also note that the output of gate 5 is $\bar{A} + B + C_1$ (which equals $A \cdot \bar{B} \cdot \bar{C}_1$) since the inputs are A , B , and C_1 . Thus, gate 1 of Figure 7 has the same output as gate 1 of Figure 8, and gate 5 of Figures 7 and 8 also have the same output. The same situation is true for gates 2, 3, 6, 7 and 8 of Figures 7 and 8. The output of gate 4 of Figure 7 (C_0) is the same as the output of gate 4 of Figure 8 when it is inverted. Gate 9 of Figure 7 provides the output (S) of the adder circuit, which is the same as the inverted output of gate 9 in Figure 8. Figures 7 and 8 also illustrate full adder logic arrangements equivalent to the connection of two half adders. The truth table and output expressions are identical to those already described. The level-by-level development of a binary addition is also shown in Figure 8. You are advised to follow the development of each digit throughout the circuit in order to further your understanding of its operation.

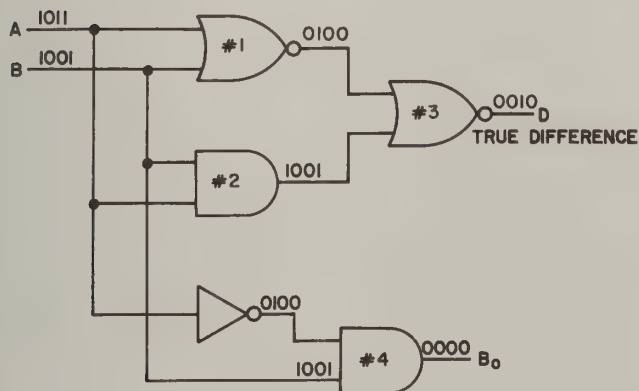
Subtractor Logic



HALF SUBTRACTOR LOGIC

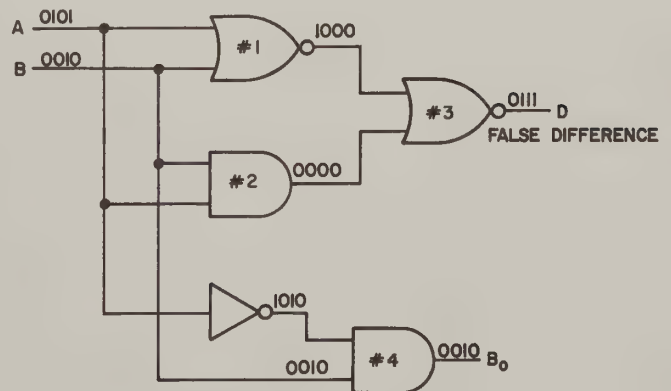
Figure 9

Figure 9 illustrates the arrangement known as the **HALF SUBTRACTOR**. The **EXCLUSIVE-OR** function ($A \cdot \bar{B} + \bar{A} \cdot B$) is available at the difference output terminal (D). A second output is available from the input of the first-level AND gate to provide the borrow condition. The arrangement causes the B_0 output to be $\bar{A} \cdot B$ (it must be 1 when A is 0 and when B is 1). Throughout this discussion, the binary number at B will be subtracted from the binary number at A. The truth data is presented in Table 2.



SUBTRACTION WITHOUT BORROW

A



SUBTRACTION WITH BORROW

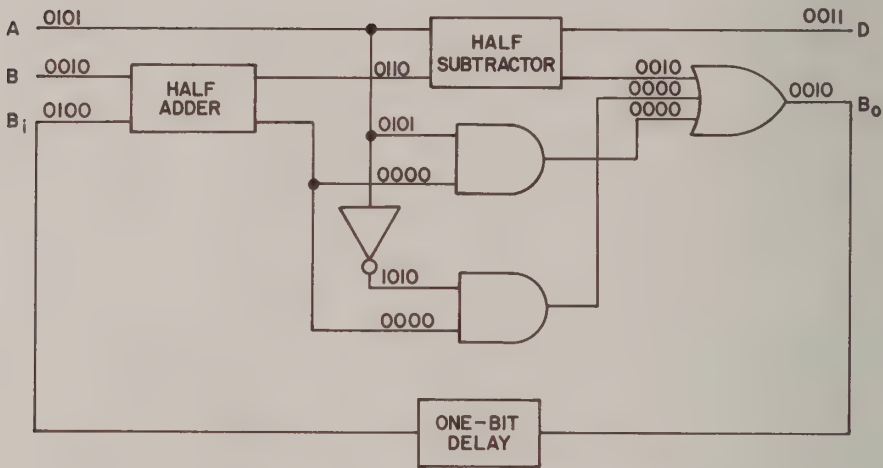
B

TRUE AND FALSE HALF SUBTRACTOR DIFFERENCE OUTPUTS

Figure 10

As with the half adder, the HALF SUBTRACTOR cannot, by itself, perform all possible binary subtractions. The difference output will only be a “TRUE DIFFERENCE” as long as there are no borrow operations. Such operations will produce a “FALSE DIFFERENCE.” Examples of the true and false differences are illustrated in Figure 10. The correct difference in Figure 10B would be:

$$\begin{array}{r} 0101 \\ -0010 \\ \hline 0011 \end{array}$$



SERIAL HALF SUBTRACTOR ARRANGEMENT TO OBTAIN THE FULL SUBTRACTOR FUNCTION

Figure 11

A	B	B _i	D	B _o
1	1	1	1	1
1	1	0	0	0
1	0	1	0	0
1	0	0	1	0
0	1	0	1	1
0	0	1	1	1
0	1	1	0	1
0	0	0	0	0

$$\begin{aligned} ABB_i + A\bar{B}\bar{B}_i + \bar{A}BB_i + \bar{A}\bar{B}\bar{B}_i &= D \\ A \cdot B + \bar{A}B_i + \bar{B}B_i &= B_o \end{aligned}$$

Truth Data for Full Subtractor

Table 5

The FULL SUBTRACTOR must have provisions for three inputs—two to handle the two binary numbers (one being subtracted from the other), the third to provide the borrow digits. A half adder and half subtractor may be arranged in serial fashion (Figure 11) to form a FULL SUBTRACTOR. The truth table for this arrangement is shown as Table 5. The truth table is developed, from Figures 4 and 11 and from Tables 1 and 2, in much the same manner as Table 3 was developed for the serial full adder. Using the combinations from Table 5 which provide a difference output of 1, the full subtractor difference sum of products expression is:

$$D = \bar{A} \cdot \bar{B} \cdot B_i + \bar{A} \cdot B \cdot \bar{B}_i + A \cdot \bar{B} \cdot \bar{B}_i + A \cdot B \cdot B_i. \tag{5}$$

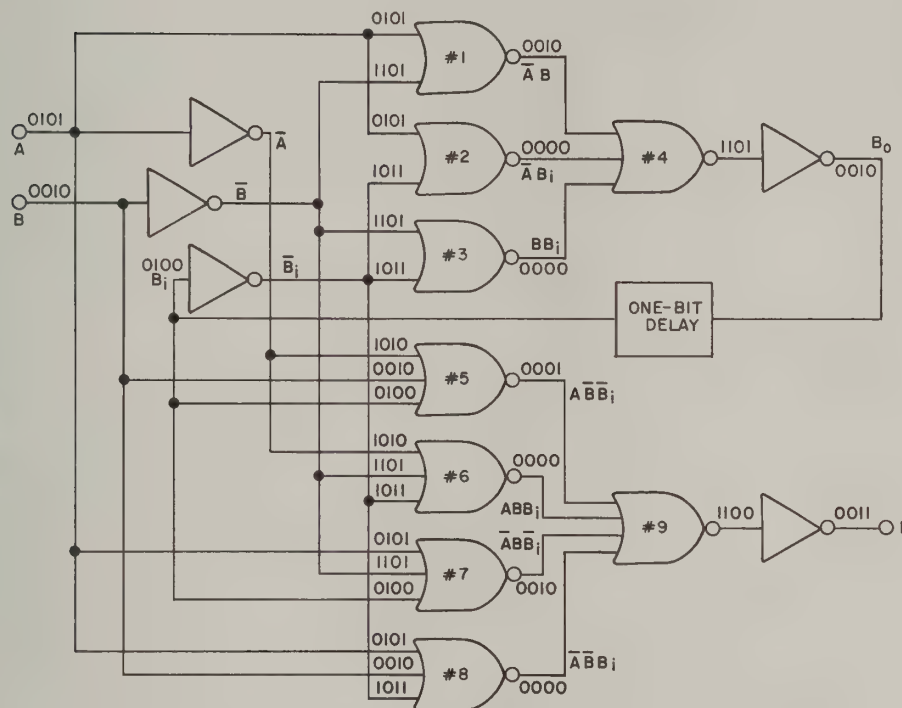
Except for the fact that B_i replaces C_i, this is identical to the sum expression, Equation 3, for the full adder (both are three-input EXCLUSIVE-OR

functions). Using those combinations from Table 5 which provide a borrow output of 1, the full subtractor borrow output expression is:

$$B_0 = \bar{A} \cdot \bar{B} \cdot B_i + \bar{A} \cdot B \cdot \bar{B}_i + \bar{A} \cdot B \cdot B_i + A \cdot B \cdot B_i,$$

which reduces to:

$$B_0 = \bar{A} \cdot B + \bar{A}B_i + BB_i. \quad (6)$$



AN ALTERNATE LOGIC ARRANGEMENT FOR THE FULL SUBTRACTOR

Figure 12

Figure 12 presents an alternate arrangement for the full subtractor. The lower logic (the portion responsible for forming the difference output) is identical to the sum-output portion of the full adder shown in Figure 8. The borrow-output portion is a modified version of the full adder carry-output arrangement.

Figure 12 is quite similar to Figure 8 (with the NOR gates and NOT circuits). The circuit of Figure 12 results from Equations 5 and 6. Gates 1 through 4 (and the NOT circuit at the output of gate 4) are the implementation of Equation 6. For gates 5 through 9, the input and output signals in Figure 12 (the implementation of Equation 5) are the same as the respec-

tive signals in Figure 8, except that C_1 in Figure 8 has been replaced by B_i in Figure 12. Furthermore, gates 1 through 4 have the same input and output signals, respectively, in Figures 8 and 12 except for A and C_1 in Figure 8 becoming $\bar{A}$ and B_i , respectively, in Figure 12.

Complementary Subtractor Logic

Complementary subtraction can be performed with a modification of the basic full adder logic arrangement. In fact, the logic can be arranged in such a fashion that, with appropriate triggering of certain gates, both addition and subtraction can be performed. The logic arrangement is shown in Figure 13. The input circuitry for B (the subtrahend) is known as the **COMPLEMENTING ARRANGEMENT**. It will apply either the true value of B or the complement of B to the full adder, depending on the trigger that is applied.

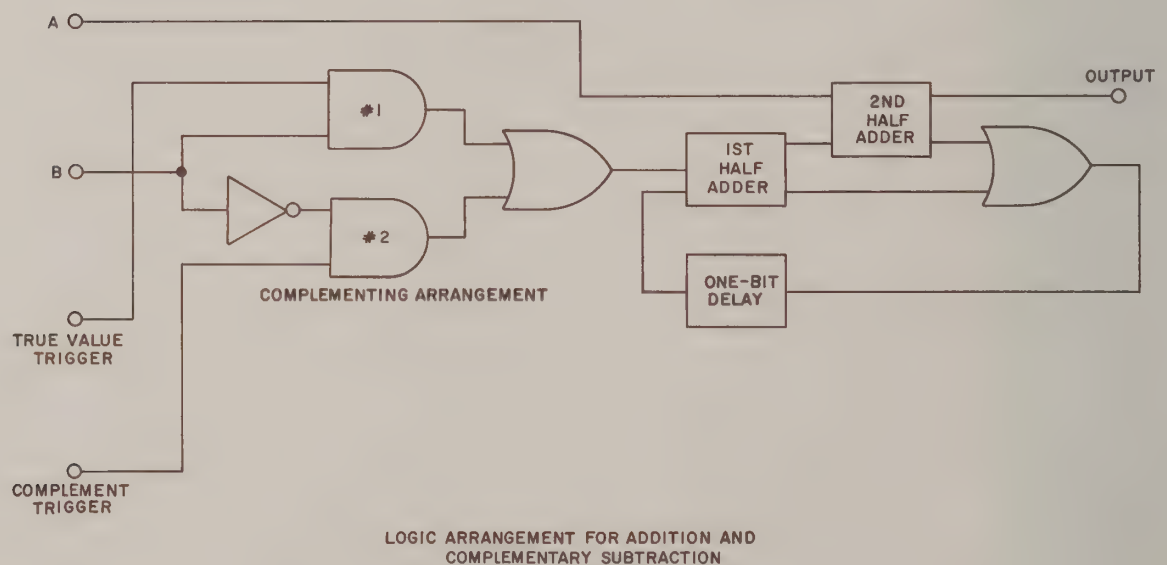


Figure 13

Operation as an adder is shown in Figure 14A. There is a logic 1 at the True Value Input and a logic 0 at the Complement Trigger. Here, the true value gate (AND gate 1) is enabled so that the input at B can pass directly to the full adder. The adder operation is identical to that shown in Figures 6 and 8.

Subtraction (through complementary addition) is shown in Figure 14B. The True Value Trigger has a logic 0, while the Complement Trigger has a logic

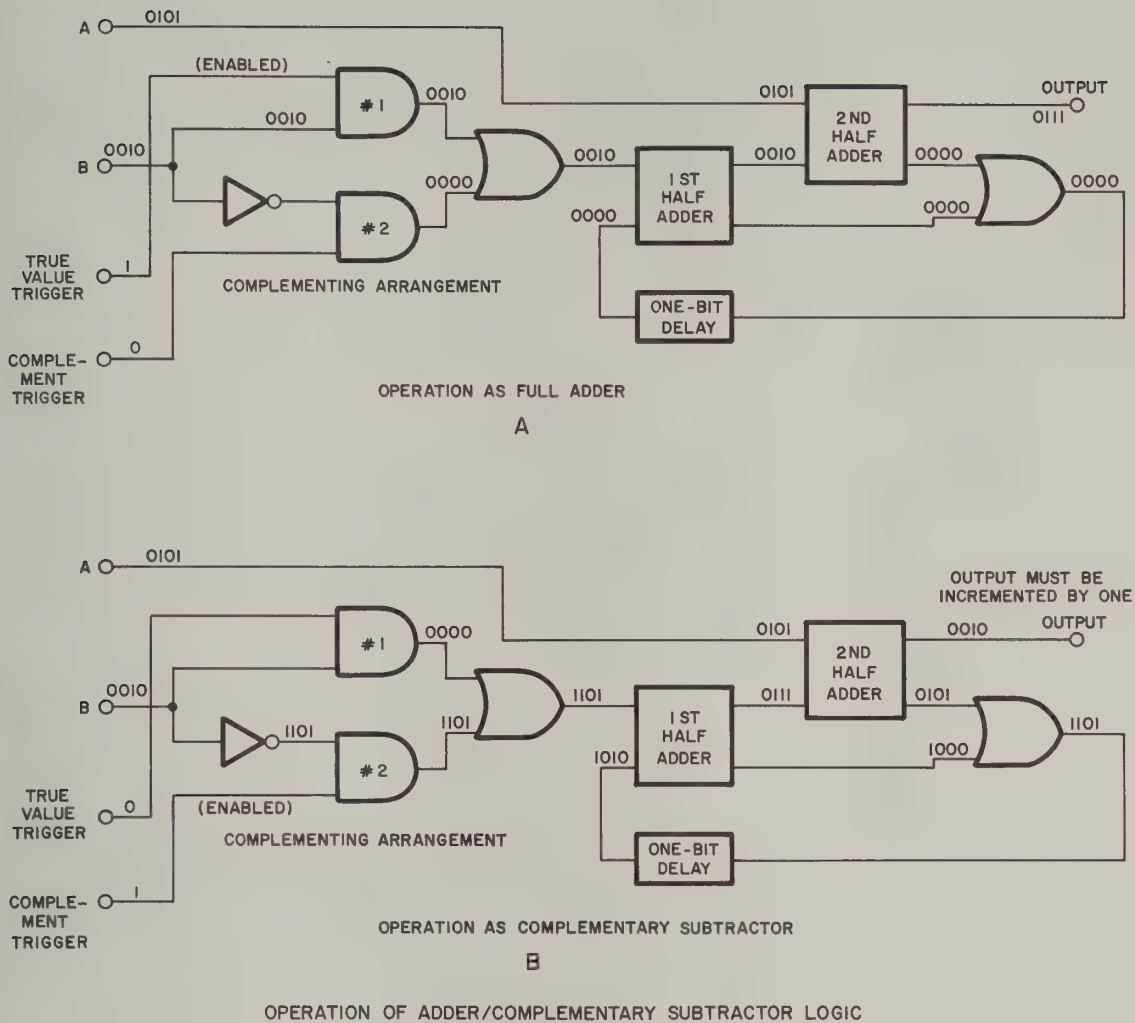


Figure 14

1. The complement gate is enabled (AND gate 2), providing the complement of B at the full adder input. (It is assumed that the subtrahend will always be input at B.) The addition operation is performed in the normal manner (similar to that of Figures 6 and 8). The next step in the "subtraction" process is to initiate the end-around carry. The complexity associated with performing this step depends on the type of numbers being processed. If the subtrahend will never be larger than the minuend, the end-around carry will always apply. Since this is simply the addition of one to the adder sum, the sum output can be temporarily stored. In the presence of the complementing trigger at B, the stored result can be incremented by one. Conversely, if the subtrahend is larger than the minuend, this incrementation must not occur.

BINARY MULTIPLICATION AND DIVISION

These two processes are actually simpler than decimal multiplication and division. Multiplication is accomplished by repeated additions, and division is performed by successive subtractions.

Multiplication

When we multiply 4 by 5 we are actually adding four 5's together:

$$4 \times 5 = 5 + 5 + 5 + 5 = 20$$

However, to make our everyday computation faster, we usually use multiplication tables that we memorized in our grade school days. Knowing these tables, we can immediately write $4 \times 5 = 20$ and $3 \times 7 = 21$. In this last example, the 7 is called the multiplier, the 3 is called the multiplicand, and 21 is called the product. Actually, in multiplication, it does not matter which number is the multiplicand and which is the multiplier.

The next step is multiplication by a two-digit number; for example, 15×12 . To accomplish correct multiplication of the 15 by the multiplier 12, we must account for the fact that each of the digits in this two-digit number has a different value, due to its relative position in the number. The right-hand digit represents 2×10^0 , or 2; while the left-hand digit represents 1×10^1 , or 10. Therefore, when multiplying by the 1 we must indicate this difference by the position in which this portion of the product (called a partial product) is written. We shift this partial product one place to the left to indicate that it is multiplied by 10. Thus:

$$\begin{array}{r} 15 \\ \times 12 \\ \hline 30 \\ 15 \\ \hline 180 \end{array}$$

Due to its position, the number 15 in the partial product represents 150 and not 15, because we actually multiply 15×10^1 or 15×10 , producing a partial product of 150. Adding the partial products, we have $150 + 30 = 180$. Thus, shifting the 15 one place to the left multiplied it by 10. As another example, multiply 456×1021 :

$$\begin{array}{r} 456 \\ \times 1021 \\ \hline 456 \\ 912 \\ 456 \\ \hline 465576 \end{array}$$

The first partial product is 456×1 . Notice that 912 is shifted one position to the left to account for the fact that it is the product of 456×20 and not 456×2 . Also, the last 456 is shifted three positions to the left to indicate that it is the product of 456×1000 and not 456×1 . These same considerations hold true for binary numbers as well as decimal numbers. Addition can be used but the use of multiplication tables and the shifting of partial products makes the computation much faster. Fortunately, the multiplication equations for single-digit binary numbers are extremely simple:

$$0 \times 0 = 0 \quad (7a)$$

$$0 \times 1 = 0 \quad (7b)$$

$$1 \times 0 = 0 \quad (7c)$$

$$1 \times 1 = 1 \quad (7d)$$

Also, for binary numbers, shifting to the left to the next digit column no longer multiplies by 10; it multiplies by 2. However, the mechanics of the procedure are exactly the same. For example:

$$\begin{array}{r} 101 \\ \times 100 \\ \hline 10100 \end{array} = \begin{array}{r} 5 \\ \times 4 \\ \hline 20 \end{array}$$

Notice that multiplying by 100 binary (4 decimal) merely shifts the multiplicand 101 two places left for the product. As another example:

$$\begin{array}{r} 1001 \\ \times 101 \\ \hline 1001 \\ 1001 \\ \hline 1001 \\ 101101 \end{array} = \begin{array}{r} 9 \\ \times 5 \\ \hline 45 \end{array}$$

Notice that the second partial product is shifted two places to the left to indicate that 1001 is multiplied by 100 binary (4 decimal) and not by 1. And, as a third example:

$$\begin{array}{r} 1000101 \\ \times 110101 \\ \hline 1000101 \\ 1000101 \\ 1000101 \\ 1000101 \\ 1000101 \\ \hline 111001001001 \end{array} = \begin{array}{r} 69 \\ \times 53 \\ \hline 207 \\ 345 \\ 3657 \end{array}$$

Notice from this third example that binary multiplication is nothing more than writing the multiplicand down for each time a 1 appears in the multiplier. However, each time the multiplicand is shifted it is multiplied by the same power of 2 as the bit in the multiplier. Consequently, computers using

binary numbers can be designed to multiply simply by using the regular adders in conjunction with a shift and add-in mechanism which is controlled by the 1's in the multiplier.

Division

The division of binary numbers is the same as decimal division except that it is much simpler. In the decimal system you not only determine whether a number is larger than the divisor but, if it is larger, how many times larger. In the binary system it is only necessary to determine if the number is as large as or larger than the divisor. If the number is larger, a 1 is placed in the quotient; if not, a 0 is placed in the quotient. The rules for dividing one bit by another are:

$$0 \div 1 = 0 \quad (8a)$$

$$1 \div 1 = 1 \quad (8b)$$

As in the decimal system (or in any system), division by zero, such as $0 \div 0$ or $1 \div 0$, is meaningless. That is, the quotient obtained when any number is divided by 0 is indeterminate. The following examples show the similarity between binary and decimal division:

$$\begin{array}{r} 101 \\ 100 \overline{)10100} \\ \underline{100} \\ 100 \\ \underline{100} \\ 0 \end{array} = \begin{array}{r} 5 \\ 4 \overline{)20} \\ \underline{20} \\ 0 \end{array}$$

$$\begin{array}{r} 10010 \\ 110 \overline{)1101100} \\ \underline{110} \\ 110 \\ \underline{110} \\ 0 \end{array} = \begin{array}{r} 18 \\ 6 \overline{)108} \\ \underline{6} \\ 48 \\ \underline{48} \\ 0 \end{array}$$

$$\begin{array}{r} 1111 \\ 111 \overline{)1101001} \\ \underline{111} \\ 1100 \\ \underline{111} \\ 1010 \\ \underline{111} \\ 111 \\ \underline{111} \\ 0 \end{array} = \begin{array}{r} 15 \\ 7 \overline{)105} \\ \underline{7} \\ 35 \\ \underline{35} \\ 0 \end{array}$$

$$\begin{array}{r}
 1100 \\
 111 \overline{) 1010100} \\
 \underline{111} \\
 111 \\
 \underline{111} \\
 0
 \end{array}
 =
 \begin{array}{r}
 12 \\
 7 \overline{) 84} \\
 \underline{7} \\
 14 \\
 \underline{14} \\
 0
 \end{array}$$

$$\begin{array}{r}
 1100 \\
 111 \overline{) 1010111} \\
 \underline{111} \\
 111 \\
 \underline{111} \\
 11
 \end{array}
 =
 \begin{array}{r}
 12 \\
 7 \overline{) 87} \\
 \underline{7} \\
 17 \\
 \underline{14} \\
 3
 \end{array}$$

In the last decimal and binary example, the answers can be written as “12 and a remainder of 3” or “ $12\frac{3}{7}$ ” for the decimal portion of the example; and “1100 and a remainder of 11” or “ $1100\frac{11}{111}$ ” for the binary portion of the example.

It might be well to point out that computers do not inspect and thus determine how many times the divisor can be subtracted. Instead, they keep subtracting the divisor until the remainder is less than the divisor. A counter which indicates the number of subtractions gives the quotient. An alternate technique consists of using complementary numbers, which are easily set by inverting the register's states. Then the subtractions can be replaced by additions, thus making the adder the basic unit for all arithmetic operations.

SUMMARY

Binary addition and subtraction involve the same procedures that are associated with the addition and subtraction of decimal values. These procedures include a set of basic sums and differences, a carry process and a borrow process. The decimal addition and subtraction “facts” involve the memorization of many digit combinations; the binary operations only involve eight basic combinations.

In the binary number system, negative values may be indicated by adding a “sign bit” to the left-most end of the code group. A second method is to convert the value into its complemented form. Two types of complementary numbers are used. The true complement is obtained by subtracting each digit in the number from the maximum digit value in the number system, then adding one to the right-most digit in the result. The radi-minus-one complement is obtained by subtracting each digit in the number from the maximum digit value in the number system. In the binary system, the true

complement is known as the 2's complement; the radix-minus-one complement, as the 1's complement.

The subtraction process can be converted to an addition process if the subtrahend is converted to its complementary value. Either type complement may be used. However, the specific rules for performing the operation depend on which type complement is used. The 1's complement is especially useful in digital systems because it can be obtained by inverting the given binary digits.

The EXCLUSIVE-OR logic function is used as a basis for performing the addition and subtraction operations. For addition, the circuit must be modified to include the carry operation (which is basically an AND function). For subtraction, the circuit must be able to perform the borrow operation (which is basically an INHIBIT function).

The half adder is a two-input, modified EXCLUSIVE-OR arrangement. It can perform a true addition only if the operation does not involve a carry. The full adder can be a combination of two half adders, modified to enable the carry output of one stage to be delayed and fed back as a third input to the first stage. The half subtractor is also a two-input logic arrangement. It can perform true subtraction only if there is no borrow operation. The full subtractor can be a combination of two half subtractors, modified to perform the borrow operation.

For complementary subtraction, the basic EXCLUSIVE-OR arrangement is also used. However, the subtrahend input includes an inverter if the 1's complement is to be used. Also, the output must be placed in a register and incremented by one if a carry occurs in the left-most addition column. The same circuitry can be used for addition and subtraction, provided that the complementing circuitry can be triggered to provide either the true value or the complementary value.

Binary multiplication is a process of repeated addition. Conversely, binary division is a process of repeated subtraction. However, when these operations are implemented in a digital system, the process is simplified by shifting the binary number code. Each left shift multiplies the number by 2. Each right shift divides the number by 2.

The following Practice Exercise questions cover the subjects which you have just studied. They are:

ARITHMETIC LOGIC

ADDER LOGIC

SUBTRACTOR LOGIC

COMPLEMENTARY SUBTRACTOR LOGIC

BINARY MULTIPLICATION AND DIVISION

MULTIPLICATION

DIVISION

14. Both binary addition and binary subtraction are basically _____ functions.
15. The carry operation in binary addition is the basic _____ function.
16. The _____ function is used to implement the borrow operation in binary subtraction.
17. Obtaining the 1's complement for a binary number simply requires the use of the _____ function.
18. The half adder is an EXCLUSIVE-OR network with an additional output to indicate the carry operation. True or False?
19. Justify the carry output of row two in Table 1.
20. Why does the half adder produce a "false sum" for certain binary number combinations?
21. Justify the first digit (right-most digit) outputs in Figure 6 on a level-by-level basis.
22. Trace the development of both outputs for 0111 at input A and 0011 at input B (decimals 7 and 3), based on a level-by-level tracing of the conditions through the network in Figure 8.
23. The difference and sum operations are both identical EXCLUSIVE-OR operations, except for the carry and borrow provisions. True or False?
24. Justify the borrow output in row three of Table 2.

25. The half subtractor will always provide a “true difference” output. True or False?
26. Explain the fifth-line difference and borrow outputs in Table 5 by tracing the conditions through the network in Figure 11.
27. Explain the operation of the full subtractor in Figure 12 by tracing the conditions when 00100 is subtracted from 01011.
28. By observing the operation of the logic arrangement in Figure 14, how can you tell whether an addition or a subtraction is being performed?
29. The end-around carry operation is equivalent to adding _____ to the adder sum, but this would be performed only in the presence of the _____ trigger.
30. Perform the following binary multiplications:
- (a)
$$\begin{array}{r} 101 \\ \times 11001 \\ \hline \end{array}$$
- (b)
$$\begin{array}{r} 10001 \\ \times 11010 \\ \hline \end{array}$$
- (c)
$$\begin{array}{r} 1001 \\ \times 101 \\ \hline \end{array}$$
31. Convert the values in Practice Exercise 30 into decimal numbers and perform the decimal multiplications.
32. Perform the following binary divisions:
- (a) $1000110 \div 101$, (b) $10111101 \div 1001$, (c) $11001 \div 1001$,
(d) $110001 \div 111$.
33. Convert the values in Practice Exercise 32 into decimal numbers and perform the decimal divisions.
34. To double a binary number in a computer, we can simply shift the number code one place to the (left/right).
35. To divide a binary number in a computer by binary 64, we can simply shift the number code _____ places to the _____.

IMPORTANT DEFINITIONS

BORROW—When one digit of a number is subtracted from a smaller digit in another number, the act of going to the next digit to the left in the minuend, reducing its value by one and adding the radix to the column being solved. Also, the digits borrowed from the next column.

CARRY—When a column of digits is added, the act of writing down the right-most digits in the sum and carrying the remaining digits over to the next columns. Also, the digits carried over to the next column.

COMPLEMENTED FORM—A number form used to represent negative numbers. The two main types of complements are: the true complement and the radix-minus-one complement. The **TRUE COMPLEMENT** is called the 10's complement in the decimal system and the 2's complement in the binary system. The **RADIX-MINUS-ONE COMPLEMENT** is called the 9's complement in the decimal system and the 1's complement in the binary system.

COMPLEMENTING ARRANGEMENT—A logic arrangement associated with adders used to perform both addition and complementary subtraction. Depending on the trigger applied, the arrangement will pass either the true value of the input or the 1's complement of the input.

FALSE DIFFERENCE—A false numerical result obtained when the subtraction of one binary number from another binary number by a **HALF SUBTRACTOR** involves a borrow operation.

FALSE SUM—A false numerical result obtained when the addition of two binary numbers by a **HALF ADDER** involves a carry operation.

FULL ADDER—A logical **EXCLUSIVE-OR** network for adding binary numbers with provisions for a carry input. These adders produce a **TRUE SUM** for all binary numbers.

FULL SUBTRACTOR—A logical **EXCLUSIVE-OR** network for subtracting binary numbers with provisions for a borrow input. These subtractors produce a **TRUE DIFFERENCE** for all binary numbers.

HALF ADDER—A logical **EXCLUSIVE-OR** network for adding binary numbers without provisions for a carry input. When the numbers being added involve a carry operation, the result is a **FALSE SUM**.

HALF SUBTRACTOR—A logical **EXCLUSIVE-OR** network for subtracting binary numbers without provisions for a borrow input. When the subtraction involves a borrow operation, the result is a **FALSE DIFFERENCE**.

IMPORTANT DEFINITIONS (Continued)

RADIX-MINUS-ONE COMPLEMENT—See COMPLEMENTED FORM.

REGISTER—An arrangement of flip-flops used to temporarily store a set of binary digits.

TRUE COMPLEMENT—See COMPLEMENTED FORM.

TRUE DIFFERENCE—A true numerical result obtained when one binary number is subtracted from another binary number.

TRUE SUM—A true numerical result obtained when two binary numbers are added.

ESSENTIAL SYMBOLS AND EQUATIONS

A, B	logical inputs, general
S	sum output, adder logic
C_o	carry output, adder logic
C_i	carry input, adder logic
D	difference output, subtractor logic
B_o	borrow output, subtractor logic
B_i	borrow input, subtractor logic

$$0 + 0 = 0 \quad (1a)$$

$$0 + 1 = 1 \quad (1b)$$

$$1 + 0 = 1 \quad (1c)$$

$$1 + 1 = 0 \text{ and a carry of } 1 \quad (1d)$$

$$0 - 0 = 0 \quad (2a)$$

$$1 - 0 = 1 \quad (2b)$$

$$1 - 1 = 0 \quad (2c)$$

$$0 - 1 = 1 \text{ with a borrow of } 1 \quad (2d)$$

$$S = \bar{A} \cdot \bar{B} \cdot C_i + \bar{A} \cdot B \cdot \bar{C}_i + A \cdot \bar{B} \cdot \bar{C}_i + A \cdot B \cdot C_i \quad (3)$$

$$C_o = A \cdot B + A \cdot C_i + B \cdot C_i \quad (4)$$

$$D = \bar{A} \cdot \bar{B} \cdot B_i + \bar{A} \cdot B \cdot \bar{B}_i + A \cdot \bar{B} \cdot \bar{B}_i + A \cdot B \cdot B_i \quad (5)$$

$$B_o = \bar{A} \cdot B + \bar{A}B_i + BB_i \quad (6)$$

$$0 \times 0 = 0 \quad (7a)$$

$$0 \times 1 = 0 \quad (7b)$$

$$1 \times 0 = 0 \quad (7c)$$

$$1 \times 1 = 1 \quad (7d)$$

$$0 \div 1 = 0 \quad (8a)$$

$$1 \div 1 = 1 \quad (8b)$$

PRACTICE EXERCISE SOLUTIONS

1. True

2. (a) 00110

(b) 01011

(c) 01111

(d) 11 (Carry)

$$\begin{array}{r} 00011 \\ +00110 \\ \hline 01001 \end{array}$$

(e) 1 (Carry)

$$\begin{array}{r} 01010 \\ +00011 \\ \hline 01101 \end{array}$$

(f) 111 (Carry)

$$\begin{array}{r} 00111 \\ +00011 \\ \hline 01010 \end{array}$$

3. (a)

$$\begin{array}{r} 2 \\ +4 \\ \hline 6 \end{array}$$

(b)

$$\begin{array}{r} 3 \\ +8 \\ \hline 11 \end{array}$$

(c)

$$\begin{array}{r} 6 \\ +9 \\ \hline 15 \end{array}$$

(d)

$$\begin{array}{r} 3 \\ +6 \\ \hline 9 \end{array}$$

(e)

$$\begin{array}{r} 10 \\ +3 \\ \hline 13 \end{array}$$

(f)

$$\begin{array}{r} 7 \\ +3 \\ \hline 10 \end{array}$$

4. (a) 00011

(b) 00001

(c) 10000

(d) 01 (Borrow)

$$\begin{array}{r} 01100 \\ -01001 \\ \hline 00011 \end{array}$$

(e) 011 (Borrow)

$$\begin{array}{r} 10000 \\ -00010 \\ \hline 01110 \end{array}$$

(f) 01 (Borrow)

$$\begin{array}{r} 01001 \\ -00111 \\ \hline 00010 \end{array}$$

5. (a)

$$\begin{array}{r} 15 \\ -12 \\ \hline 3 \end{array}$$

(b)

$$\begin{array}{r} 9 \\ -8 \\ \hline 1 \end{array}$$

(c)

$$\begin{array}{r} 19 \\ -3 \\ \hline 16 \end{array}$$

(d)

$$\begin{array}{r} 12 \\ -9 \\ \hline 3 \end{array}$$

(e)

$$\begin{array}{r} 16 \\ -2 \\ \hline 14 \end{array}$$

(f)

$$\begin{array}{r} 9 \\ -7 \\ \hline 2 \end{array}$$

6. False—The extra bit is placed at the left-hand end of a binary code group.

7. complemented

8. The two types of complemented forms are the true complement and the radix-minus-one complement.

9. (a)

$$\begin{array}{r} 11111 \\ -01100 \\ \hline 10011 \\ +1 \\ \hline 10100 \end{array}$$

(b)

$$\begin{array}{r} 11111 \\ -01000 \\ \hline 10111 \\ +1 \\ \hline 11000 \end{array}$$

(c)

$$\begin{array}{r} 11111 \\ -00011 \\ \hline 11100 \\ +1 \\ \hline 11101 \end{array}$$

PRACTICE EXERCISE SOLUTIONS (Continued)

$$\begin{array}{r} \text{(d)} \quad 11111 \\ -01001 \\ \hline 10110 \\ + \quad 1 \\ \hline 10111 \end{array}$$

$$\begin{array}{r} \text{(e)} \quad 11111 \\ -00010 \\ \hline 11101 \\ + \quad 1 \\ \hline 11110 \end{array}$$

$$\begin{array}{r} \text{(f)} \quad 11111 \\ -00111 \\ \hline 11000 \\ + \quad 1 \\ \hline 11001 \end{array}$$

$$10. \text{ (a)} \quad 10011$$

$$\text{(b)} \quad 10111$$

$$\text{(c)} \quad 11100$$

$$\text{(d)} \quad 10110$$

$$\text{(e)} \quad 11101$$

$$\text{(f)} \quad 11000$$

$$11. \text{ (a)} \quad 01111$$

$$\begin{array}{r} + 10100 \\ \hline 100011 \\ \downarrow \\ \text{Drop the Carry} \end{array} = 00011$$

$$\text{(b)} \quad 01001$$

$$\begin{array}{r} + 11000 \\ \hline 100001 \\ \downarrow \\ \text{Drop the Carry} \end{array} = 00001$$

$$\text{(c)} \quad 10011$$

$$\begin{array}{r} + 11101 \\ \hline 110000 \\ \downarrow \\ \text{Drop the Carry} \end{array} = 10000$$

$$12. \text{ (d)} \quad 01100$$

$$\begin{array}{r} + 10110 \\ \hline 100010 \text{ (End-Around Carry)} \\ \downarrow \rightarrow 1 \\ \hline 00011 \end{array}$$

$$\text{(e)} \quad 10000$$

$$\begin{array}{r} + 11101 \\ \hline 101101 \text{ (End-Around Carry)} \\ \downarrow \rightarrow 1 \\ \hline 01110 \end{array}$$

$$\text{(f)} \quad 01001$$

$$\begin{array}{r} + 11000 \\ \hline 100001 \text{ (End-Around Carry)} \\ \downarrow \rightarrow 1 \\ \hline 00010 \end{array}$$

$$13. \text{ (a)} \quad 10111$$

$$\begin{array}{r} -11000 \\ \hline -00001 \end{array}$$

Rule 1: Find radix-minus-one of the subtrahend (1's complement):

$$00111$$

Rule 2: Add the complement to the minuend:

$$\begin{array}{r} 10111 \\ +00111 \\ \hline 11110 \end{array}$$

Rule 4: If the addition of the left column does not develop a carry, obtain the radix-minus-one complement of the result. The complement is the desired answer and it is negative:

$$-00001$$

PRACTICE EXERCISE SOLUTIONS (Continued)

$$\begin{array}{r}
 \text{(b)} \quad 01010 \\
 -10101 \\
 \hline
 -01011
 \end{array}$$

Rule 1: 01010

$$\begin{array}{r}
 \text{Rule 2:} \quad 01010 \\
 +01010 \\
 \hline
 10100
 \end{array}$$

Rule 4: -01011

14. EXCLUSIVE-OR

15. AND

16. INHIBIT

17. NOT

18. True

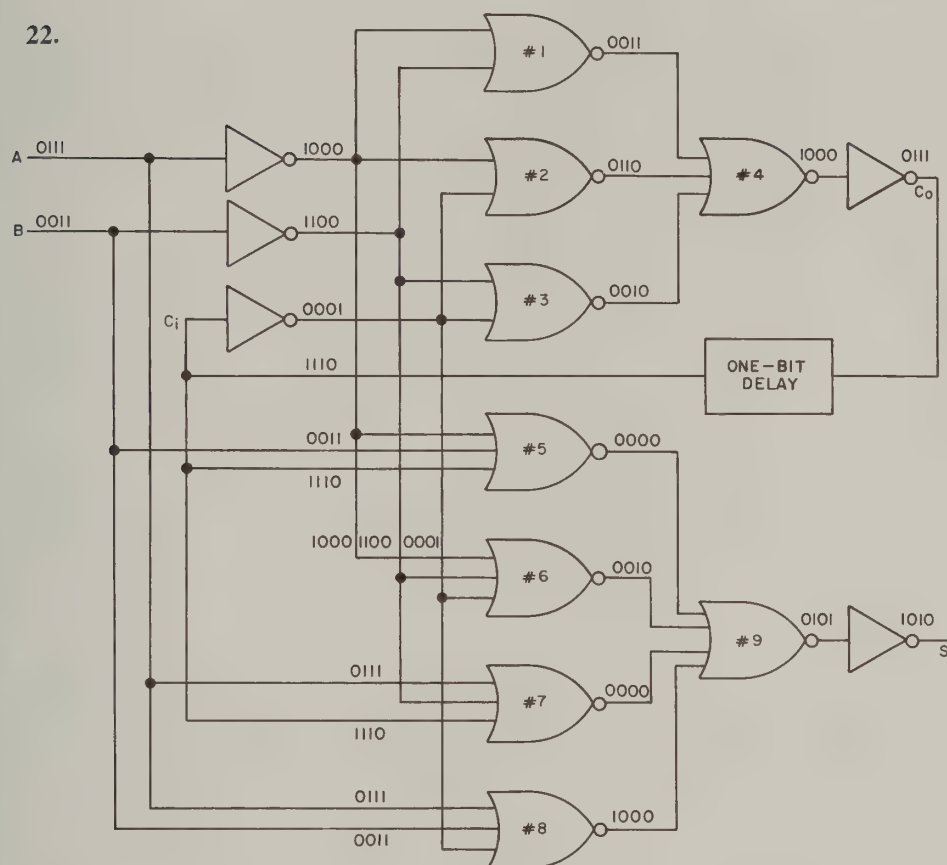
19. Row two applies when a 0 and 1 are being added. This addition has no carry. Hence, the carry output is 0.

20. When two binary 1's are added, the sum output of the half adder is 0, leading to a "false sum."

21. Two 1's are being added with a 0 carry input. With a 1 and 0 input to the first half adder, the sum output is 1 and the carry output is 0 (from Table 1). With two 1's applied to the second half adder, its sum output (the final sum output) is 0 and its carry output is 1 (also from Table 1). The first-level and second-level carry outputs (0 and 1, respectively) are applied to an OR gate, producing a 1 for the final carry output.

PRACTICE EXERCISE SOLUTIONS (Continued)

22.



23. True

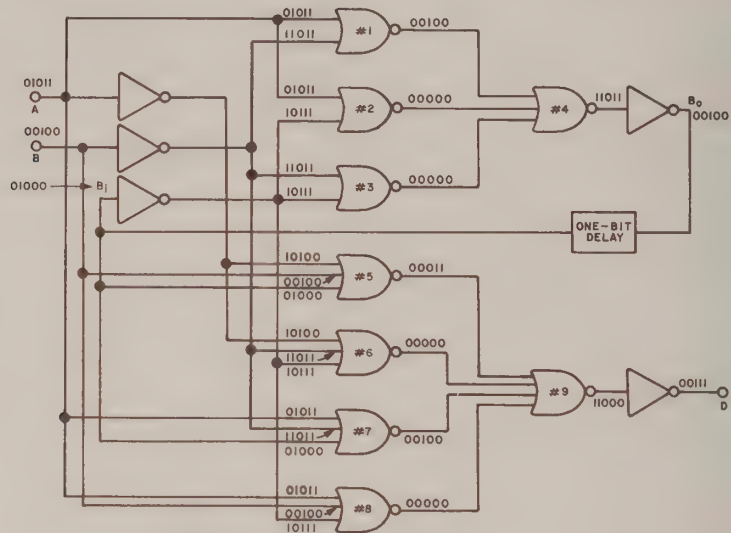
24. Row three applies when 0 is being subtracted from 1. This subtraction has no borrow. Hence, the borrow output is 0.

25. False—A false difference results if a borrow operation occurs.

26. Initial conditions from Table 5, line 5: $A = 0$, $B = 1$, $B_1 = 0$. Considering the B and B_1 inputs to the half adder of Figure 11, the half adder sum output is 1 and its carry output is 0. The carry output is sent to both AND gates as an input. The upper AND has as its other input the 0 from the A input, which yields a 0 output. The lower AND gate has the 1 output of the NOT gate from the 0 input of A ; the output of the AND is 0. The 1 from the half adder sum output is the other input, along with the 0 from A , to the half subtractor, which has a difference output of 1 and a borrow output of 1. The 1 of the borrow output and the 0's from the AND gates serve as the inputs to the OR gate which has a 1 output (B_0).

PRACTICE EXERCISE SOLUTIONS (Continued)

27.



28. Addition is being performed when the true value trigger is present; subtraction, when the complementary trigger is present.

29. one; complementary

30. (a)
$$\begin{array}{r} 101 \\ \times 11001 \\ \hline 101 \\ 10100 \\ 101 \\ \hline 1111101 \end{array}$$
 (b)
$$\begin{array}{r} 10001 \\ \times 11010 \\ \hline 100010 \\ 10001 \\ \hline 110111010 \end{array}$$
 (c)
$$\begin{array}{r} 1001 \\ \times 101 \\ \hline 1001 \\ 10010 \\ \hline 101101 \end{array}$$

31. (a)
$$\begin{array}{r} 5 \\ \times 25 \\ \hline 125 \end{array}$$
 (b)
$$\begin{array}{r} 17 \\ \times 26 \\ \hline 102 \\ 34 \\ \hline 442 \end{array}$$
 (c)
$$\begin{array}{r} 9 \\ \times 5 \\ \hline 45 \end{array}$$

32. (a)
$$\begin{array}{r} 1110 \\ 101 \overline{)1000110} \\ \underline{101} \\ 00111 \\ \underline{101} \\ 0101 \\ \underline{101} \\ 00 \end{array}$$
 (b)
$$\begin{array}{r} 10101 \\ 1001 \overline{)10111101} \\ \underline{1001} \\ 001011 \\ \underline{1001} \\ 001001 \end{array}$$

PRACTICE EXERCISE SOLUTIONS (Continued)

$$\begin{array}{r} \text{(c)} \quad \quad \quad 10 \\ 1001 \overline{)11001} \\ \underline{1001} \\ 00111 \end{array}$$

$$\begin{array}{r} \text{(d)} \quad \quad \quad 111 \\ 111 \overline{)110001} \\ \underline{111} \\ 1010 \\ \underline{111} \\ 111 \\ \underline{111} \\ 0 \end{array}$$

$$\begin{array}{r} 33. \text{ (a)} \quad \frac{14}{5 \overline{)70}} \\ \underline{5} \\ 20 \end{array} \quad \begin{array}{r} \text{(b)} \quad \frac{21}{9 \overline{)189}} \\ \underline{18} \\ 9 \end{array} \quad \begin{array}{r} \text{(c)} \quad \frac{2}{9 \overline{)25}} \\ \underline{18} \\ 7 \end{array} \quad \begin{array}{r} \text{(d)} \quad \frac{7}{7 \overline{)49}} \\ \underline{49} \\ 0 \end{array}$$

34. left

35. six; right

$$\begin{array}{r}
 110101 \\
 1011 \\
 \hline
 110101 \\
 110101 \\
 \hline
 110101 \\
 1001000111 \\
 \hline
 1001000111
 \end{array}$$

$$\begin{array}{r}
 7 \\
 \hline
 53
 \end{array}$$

$$\begin{array}{r}
 12 \\
 4 \\
 \hline
 1 \\
 53 \\
 \hline
 57
 \end{array}$$

$$\begin{array}{r}
 1101 \quad 11 \\
 101011 \\
 11101 \\
 \hline
 11101
 \end{array}$$

$$\begin{array}{r}
 4 \\
 16 \\
 \hline
 20
 \end{array}$$

$$\begin{array}{r}
 29 \\
 3 \\
 \hline
 32
 \end{array}$$

$$\begin{array}{r}
 1 \\
 2 \\
 4 \\
 16 \\
 64 \\
 \hline
 87
 \end{array}$$



1. $A = 11011 + 101$ IS -- 100000.

$$\begin{array}{r}
 11011 \\
 + 101 \\
 \hline
 0 \text{ carry } 1
 \end{array}
 \begin{array}{r}
 11011 \\
 0 \text{---carry } 1 \\
 + 101 \\
 \hline
 00
 \end{array}
 \begin{array}{r}
 11011 \\
 + 101 \\
 \hline
 000 \text{ carry } 1
 \end{array}
 \begin{array}{r}
 11011 \\
 0 \text{---carry } 1 \\
 + 101 \\
 \hline
 0000
 \end{array}
 \begin{array}{r}
 11011 \\
 0 \text{---carry } 1 \\
 + 101 \\
 \hline
 00000
 \end{array}
 \begin{array}{r}
 11011 \\
 + 101 \\
 \hline
 100000
 \end{array}$$

2. $A = 101101 - 10101$ IS -- 11000.

$$\begin{array}{r}
 101101 \\
 - 10101 \\
 \hline
 11000 \text{ borrow } 1
 \end{array}
 \begin{array}{r}
 01 \\
 101101 \\
 - 10101 \\
 \hline
 011000
 \end{array}$$

3. $A =$ THE TRUE COMPLEMENT OF 10101 IS -- 01011.

First find the 1's complement by changing all 1's to 0's and all 0's to 1's: 01010. Next, add 1 to the 1's complement

$$\begin{array}{r}
 01010 \\
 + 1 \\
 \hline
 01011
 \end{array}$$

4. $B =$ THE 1'S COMPLEMENT OF 101101 IS -- 010010.

To find the 1's complement, change all 1's to 0's and 0's to 1's: 010010.

5. $A =$ A SINGLE HALF-ADDER -- WILL PRODUCE A TRUE SUM ONLY WHEN THE SUM DOES NOT INCLUDE A CARRY.

In order to perform the addition of all possible binary numbers, the C_0 (carry over output) information must be shifted one position to the left and must be added to the S (sum) information. This can be accomplished with an adder which has provisions for three inputs; two to handle the binary numbers being added, the third to add in a delayed version of the C_0 information.

6. $B =$ TRUE DIFFERENCE AND SUM OPERATIONS -- ARE BOTH IDENTICAL EXCLUSIVE-OR FUNCTIONS, EXCEPT FOR THE CARRY AND BORROW PROVISIONS.

The expression for the true difference is: $D = \bar{A} \cdot \bar{B} \cdot B_i + \bar{A} \cdot B \cdot \bar{B}_i + A \cdot \bar{B} \cdot \bar{B}_i + A \cdot B \cdot B_i$. The expression for the true sum is: $S = \bar{A} \cdot B \cdot C_i + \bar{A} \cdot B \cdot \bar{C}_i + A \cdot \bar{B} \cdot \bar{C}_i + A \cdot B \cdot C_i$. Except for the fact that B_i replaces C_i , these are identical three-input EXCLUSIVE-OR functions.

7. $C =$ THE BINARY OPERATIONS $0100 + 0011$ AND $1010 - 0011$ -- BOTH PRODUCE A RESULT OF DECIMAL 7.

Since 0100 is a Binary Coded Decimal of the decimal number 4 and 0011 is a BCD of the decimal number 3, their sum is decimal 7:

$$\begin{array}{r}
 0100 \\
 + 0011 \\
 \hline
 0111 \text{ or decimal } 7.
 \end{array}$$

Since 1010 is a BCD of the decimal number 10 and 0011 is a BCD of the decimal number 3, their difference is decimal 7:

$$\begin{array}{r}
 1010 \\
 - 0011 \\
 \hline
 0111 \text{ or decimal } 7.
 \end{array}$$

8. $D =$ IN THE PROCESS OF COMPLEMENTARY SUBTRACTION, USING THE 1'S COMPLEMENT, THE END-AROUND CARRY IS NECESSARY WHEN -- THE SUBTRAHEND IS SMALLER THAN THE MINUEND. The complexity associated with performing the end-around carry depends on the type of numbers being processed. If the subtrahend will never be larger than the minuend, the end-around carry will always apply. Since this is simply the addition of one to the adder sum, the sum output can be temporarily stored. With a 1 at the complementing trigger and a 0 at the true value trigger, the stored result can be incremented by one.

9. $C = 110101 \times 1011$ IS -- 1001000111.

$$\begin{array}{r}
 110101 \\
 \times 1011 \\
 \hline
 110101 \\
 110101 \\
 110101 \\
 110101 \\
 \hline
 1001000111
 \end{array}$$

10. D = 1010111 ÷ 11101 IS -- 11.

$$\begin{array}{r} 11 \\ 11101 \overline{) 1010111} \\ \underline{11101} \\ 011101 \\ \underline{011101} \\ 000000 \end{array}$$

5235A



QUESTIONS

IMPORTANT—These instructions **MUST** be accurately followed to avoid loss, or errors in grading.

Indicate your answer on this sheet by filling in the box for the most correct answer to each question, as illustrated in the example below.

When all questions have been answered, place the answer card in the proper position to line up the boxes on the card with the boxes on the sheet.

Next, copy the complete lesson code into the space provided on the card, and fill in the answer boxes to correspond with those previously filled in on this sheet.

Before mailing, be certain your correct student number, name and address appear on the card.

LESSON CODE
5235A

Example: Which of the following liquids does not contain caffeine?
 (A) Orange juice. (B) Coffee.
 (C) Tea. (D) Cola.

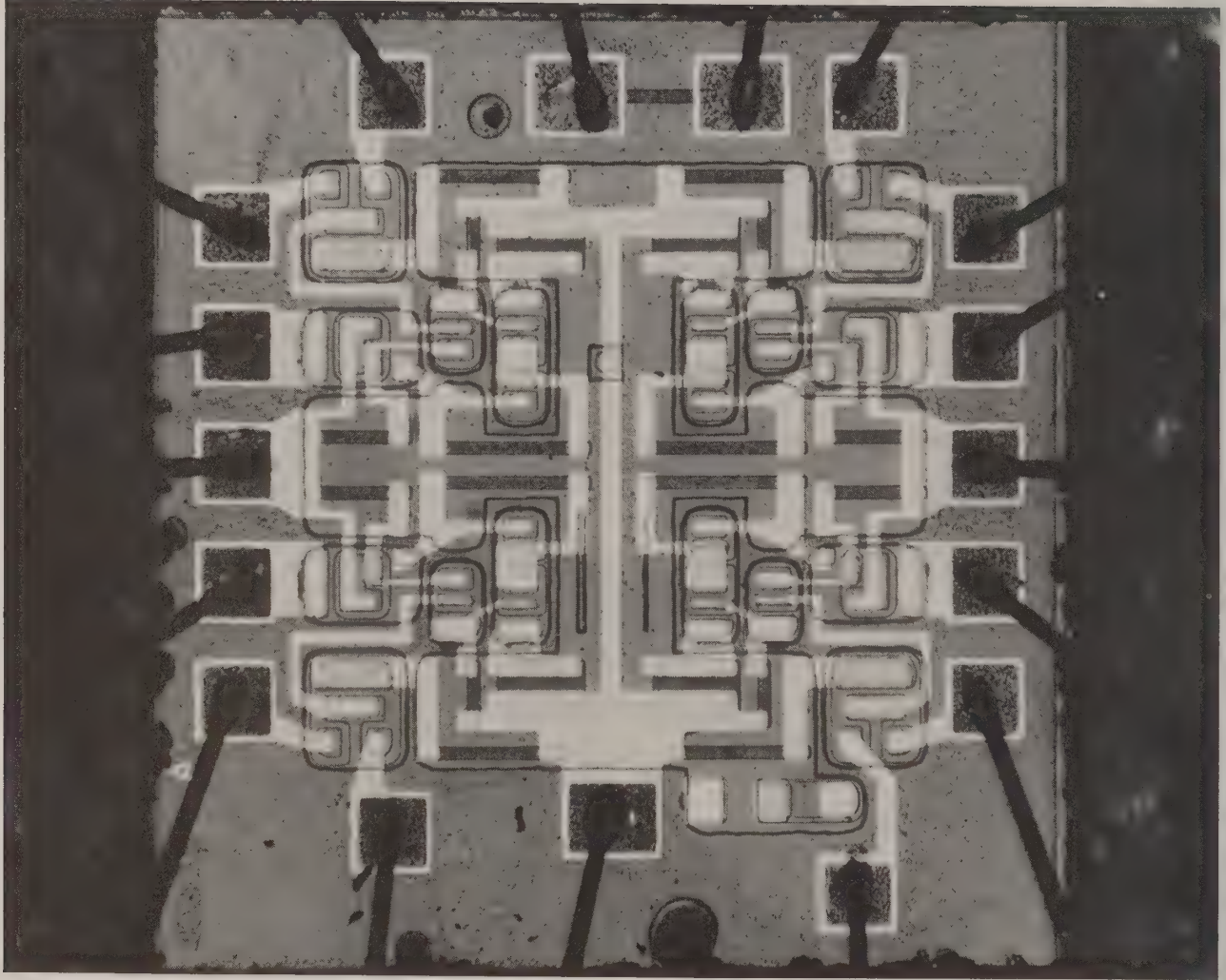
1. A ☒ 11011 + 101 is
 B ☐
 C ☐
 D ☐
 (A) 100000 (B) 11110 (C) 11100 (D) 11000
2. A ☒ 101101 - 10101 is
 B ☐
 C ☐
 D ☐
 (A) 11000 (B) 111000 (C) 101 (D) 1010
3. A ☒ The true complement of 10101 is
 B ☐
 C ☐
 D ☐
 (A) 01011 (B) 1011 (C) 01010 (D) 00010
4. A ☐ The 1's complement of 101101 is
 B ☒
 C ☐
 D ☐
 (A) 10010 (B) 010010 (C) 100010 (D) 010001
 A single half adder
 (A) will produce a true sum only when the sum does not include a carry. (B) has three inputs—two to handle the binary numbers being added; the third to provide the carry digits. (C) provides the two output expressions: $(A \cdot \bar{B}) + (\bar{A} \cdot B)$ and $A + B$ for the sum and carry outputs, respectively. (D) provides the two expressions: $\bar{A} \cdot B \cdot C_1 + \bar{A} \cdot B \cdot \bar{C}_1 + \bar{A} \cdot B \cdot C_1$ and $\bar{A} \cdot B \cdot C_1 + \bar{A} \cdot B \cdot \bar{C}_1 + A \cdot \bar{B} \cdot C_1 + A \cdot \bar{B} \cdot \bar{C}_1$ for the sum and carry outputs, respectively.
5. A ☒ True difference and sum operations
 B ☐
 C ☐
 D ☐
 (A) can be performed only over a limited range of binary numbers. (B) are both identical EXCLUSIVE-OR functions, except for the carry and borrow provisions. (C) can be accomplished with the half adder and half subtractor networks. (D) cannot be accomplished with full adder and full subtractor networks.
6. A ☐ The binary operations 0100 + 0011 and 1010 - 0011
 B ☒
 C ☐
 D ☐
 (A) are logical operations, not arithmetic operations. (B) cannot be performed in digital computers. (C) both produce a result of decimal 7. (D) can be performed with the half adder and half subtractor.
7. A ☐ In the process of complementary subtraction, using the 1's complement, the end-around carry is necessary when
 B ☐
 C ☐
 D ☒
 (A) the subtrahend is larger than the minuend. (B) there is no carry in the left column. (C) there is a carry in the right column. (D) the subtrahend is smaller than the minuend.
8. A ☐ 110101 $\times$ 1011 is
 B ☒
 C ☐
 D ☐
 (A) 101110011 (B) 1000111 (C) 1001000111 (D) 10011111
9. A ☐ 1010111 $\div$ 11101 is
 B ☐
 C ☐
 D ☒
 (A) 100 (B) 1 (C) 10 (D) 11

DIGITAL INTEGRATED CIRCUITS

5240

© COPYRIGHT 1973
BELL & HOWELL SCHOOLS, INC.
ALL RIGHTS RESERVED





The light-colored patterns on this 40 mm $\times$ 40 mm integrated circuit chip are the bands of metalization that provide the circuit interconnections. This TTL circuit has four independent, two-input NAND gates on its surface. Connections to the equipment in which it is installed are made via the .001 inch diameter wires that are attached to the squarish pads along the outside edge of the chip.

Courtesy Signetics, Inc.

CONTENTS

Integrated Circuits	Page 5
Thin-Film IC's	Page 7
Monolithic IC's	Page 8
Medium- and Large-Scale Integration	Page 11
Digital IC Types	Page 13
Resistor-Transistor Logic	Page 14
Diode-Transistor Logic	Page 15
Transistor-Transistor Logic	Page 21
Nonsaturated Logic	Page 25
Metal Oxide Semiconductor (MOS) and Complementary Metal Oxide Semiconductor (MOS) Logic	Page 28
Other IC Logic Types	Page 31
Operating Characteristics	Page 32
Temperature and Dissipation	Page 32
Voltage Swing and Noise	Page 33
Fan-In and Fan-Out	Page 34
Speed and Delay	Page 36
Comparisons of Logic Circuits	Page 36
Integrated Circuit Flip-Flops and Latches	Page 37
Set-Reset Flip-Flop	Page 38
Bistable Latches	Page 40
Type-D Flip-Flops	Page 42
Type-T Flip-Flops	Page 45
The J-K Flip-Flop	Page 47

*Some folks look so busy doing nothin' that
they seem indispensable.
—Kin Hubbard*

DIGITAL INTEGRATED CIRCUITS

Fundamentally, digital systems using integrated circuits (IC's) are no different than those using discrete components. Early in the history of IC technology, emphasis was placed on the development of digital IC's. These devices still account for the major share of IC production and sales.

Most discrete logic systems rely heavily on the use of combinations of the AND and OR functions. NOT functions are also used; but as little as possible because each is (in practice) another complete amplifier stage. The minimization of numbers of components in discrete systems is especially important when low cost is required. Since computers may require hundreds of thousands of functions, a savings of even a few cents per circuit becomes critical. Conversely, when IC's are being produced, it makes little difference whether an extra transistor or two is included in the circuit. All components are "manufactured" simultaneously in a step-by-step sequence which must be followed, regardless of the number of components, as long as there is enough physical space on the IC **CHIP**.

The NOR and NAND functions use the CE amplifier arrangement as does the NOT function. By basing digital IC design on NOR, NAND, and NOT operations, only one basic amplifier design is needed. This provides a significant reduction in pre-manufacturing costs (engineering, development, test, etc.). When needed, AND and OR operations can still be accomplished without any additional circuit design, simply by connecting between a NOR and a NOT circuit, or a NAND and a NOT circuit. This is a use of the Boolean algebra double negation process. If $\overline{A + B}$ (the NOR function) is followed by a NOT function, we have:

$$\overline{\overline{A + B}} = A + B \quad (1)$$

which is the OR function. Similarly, if $\overline{A \cdot B}$ (the NAND function) is followed by a NOT function, we have the AND function:

$$\overline{\overline{A \cdot B}} = A \cdot B. \quad (2)$$

Standard digital IC's are available for almost every conceivable logic application. Competing manufacturers are currently producing devices which are readily interchangeable. Even the packaging has become quite standardized. For simplicity, each series of basic logic circuits (regardless of the functions they perform) is commonly grouped into a **LOGIC FAMILY** or **LOGIC SYSTEM**. This grouping is based on component arrangement or operating mode, such as resistor-transistor logic (RTL), diode-transistor logic (DTL), transistor-transistor logic (TTL) or current-mode (emitter-coupled) logic (CML or ECL). Each family has its advantages and disadvantages.

The outputs of logic circuits represent logical “decisions.” Many logical decisions must be made in processing data, in controlling a machine or an industrial process, or in transmitting data from one place to another. Sometimes all of the decisions must be made simultaneously. More frequently the decisions follow sequentially in an orderly fashion. The control of such a decision process is known as timing or synchronizing. The system is then said to be **SYNCHRONOUS**. The length of time allowed for each decision is called a **BIT-TIME**. The BIT-TIME establishes the basic timing interval for the digital system. Many systems utilize equal bit-times for all operations. However, this is not essential. Many systems operate in an **ASYNCHRONOUS** mode, where a new decision sequence is initiated by a signal from the previous sequence after it indicated that it was “finished.”

Decisions in some digital systems are delayed, at least to the start of each timing interval. When longer delays are required certain conditions are stored in memory devices before the instant they influence the final signal output. Binary memory elements resemble the basic logic elements (AND, OR, NOT, NAND and NOR) because they generate logical output values of either 1 or 0. The important difference is that an input condition applied to a memory element may cause the associated output condition to persist beyond the time when the input condition is removed. The flip-flop is a type of memory device which is used in digital systems and is available in IC form. In fact, it is available in any of the IC logic families: RTL, DTL, TTL or ECL.

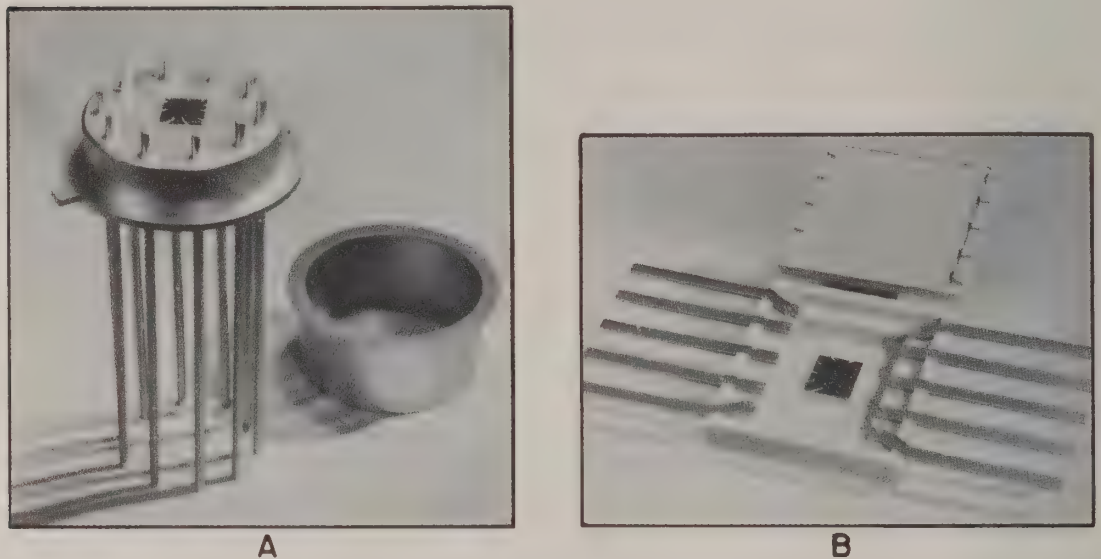
A change in the logic assignments produces the dual of the circuit. A positive logic OR circuit becomes a negative logic AND circuit; a negative logic OR circuit becomes a positive logic AND circuit; a positive logic AND circuit becomes a negative logic OR; and a negative logic AND circuit becomes a positive logic OR circuit.

INTEGRATED CIRCUITS

Transistors and integrated circuits are both made of semiconductor materials and therefore have much in common. However, there are no additional components such as resistors, diodes, capacitors, etc., within a transistor case.

Integrated circuits do contain passive and active elements within the same package. An IC is usually designed to perform a complete circuit function or a combination of complete circuit functions. IC chips, along with transistor chips, must be placed into some form of “container,” or a “package,” in order to provide adequate environmental and mechanical protection; to provide interconnection leads for tying one IC to another or to other devices; and to act as a heat sink in order to dissipate internally generated

energy. This last requirement has the greatest impact on the size of the container. The number of gates or circuits in a "package" is also limited by the number of pins or terminals that are available. Consequently, the package is usually many times larger than the chip inside.

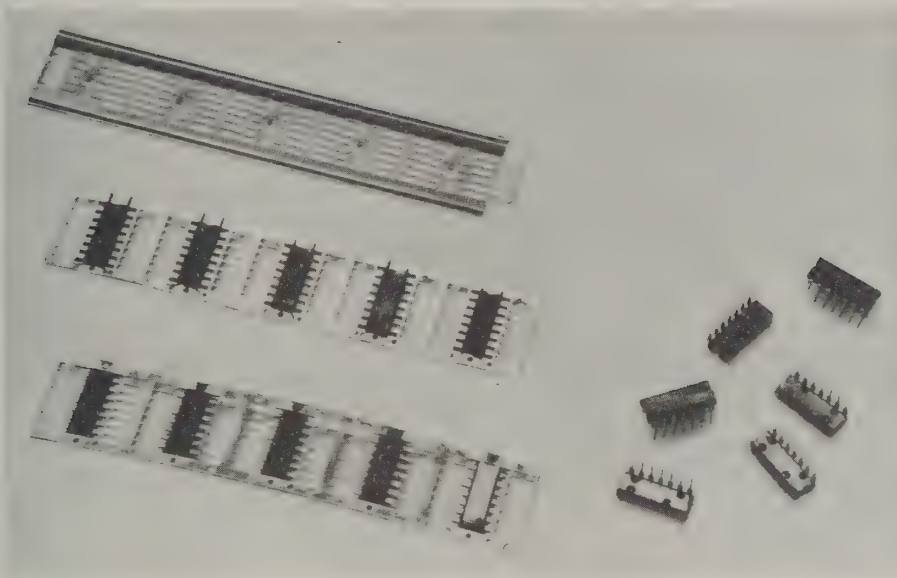


*Two Standard IC Package Types
Courtesy Hayden Book Co.*

Figure 1

Figure 1 illustrates two standard packaging approaches. The TO-type package (Figure 1A) may be a modified version of any one of several conventional transistor "cans." The primary modification consists of providing for eight, ten, twelve, or fourteen external pins. These pins protrude above the base of the container inside the can and provide contact points for very fine wires (visible in the figure), which tie to **BONDING PADS** on the IC chip. This type package is inexpensive and has a long history of reliable operation. The "flat package" in Figure 1B is composed of ceramic materials. It is considerably more expensive, and offers certain advantages over the TO package. These advantages include a larger stacking density and a more efficient use of the circuit board leads. Both of these types can be hermetically sealed.

An inexpensive standard package is composed of plastic, formed as shown in Figure 2. This approach is very adaptable to automatic mass production. Strips of the electrode patterns are machined, and the IC is bonded within the center of each pattern. The plastic encapsulation is molded around the chip, and the interlead shorts are machined away from the outer electrodes. Finally, the packages are clipped apart and the lead-frame is trimmed away.



*Plastic Packaging (Stripline Process)
Courtesy Hayden Book Co.*

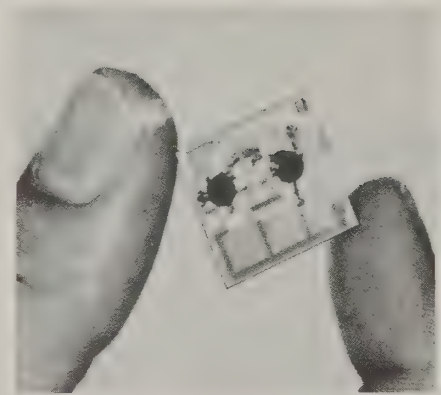
Figure 2

Plastic does not dissipate heat as well as metal or ceramic; therefore, this package is generally used for industrial applications, where the environment and circuit specifications are not overly stringent.

Thin-Film IC's

The **THIN-FILM IC** utilizes components which are less than 0.00004 inch (one micron) thick. This is accomplished by evaporating the materials onto an **INSULATING SUBSTRATE** of glass or ceramic (Figure 3). Diodes and transistors must be attached to the circuit in a discrete manner. For circuits containing a very large number of resistors and capacitors and only a few diodes and transistors, thin-film circuits are well-suited.

The electrical characteristics of thin-film components are determined by the type of material used and the geometry of the component pattern. The material selection dictates the kinds of fabrication processes which can be used, and the precautions which must be taken to maintain the desired characteristics and tolerances. Resistance accuracies of 0.1% and capacitance accuracies of better than 1% can be achieved.



*Thin-Film Circuit
Courtesy Autonetics Div.,
North American Rockwell Corp.*

*Figure
3*

Resistive materials include nickel-chromium, tin oxide, tantalum, chromium, and rhenium. This offers a wide range of material stabilities, fabrication options, and actual resistance values (from 5 ohms to 100,000 megohms). Practical film thickness ranges from 100 to 1000 angstroms ($\text{\AA}$), one $\text{\AA}$ being equal to 0.000000004 (4×10^{-9}) inch, with 300 $\text{\AA}$ to 500 $\text{\AA}$ generally used. Films thinner than 100 $\text{\AA}$ tend to be discontinuous, being broken up by the surface roughness of the substrate. Films thicker than 1000 $\text{\AA}$ cause resistances which are too low. This requires longer-length resistive components in order to achieve the necessary resistances.

Thin-film capacitors generally consist of an aluminum or gold film, topped by a dielectric film, then topped by a second film of aluminum or gold. The capacitance value is established on the basis of the dielectric material and its thickness. The exact value is obtained by adjusting the area of the component.

Dielectric materials include silicon monoxide, silicon dioxide, tantalum oxide, aluminum oxide, titanium oxide, and boroaluminum silicate. These dielectric materials provide capacitance values ranging from 0.01 picofarad/square mil to 1.6 picofarad/square mil.

Aluminum and gold are two of the most widely used conductive materials for intercomponent connections (equivalent to the "wiring" in a conventional circuit). Since these materials are also used for the lower capacitor plates, the two are usually put on the substrate in a single operation.



Monolithic Circuit Chip

Figure
4

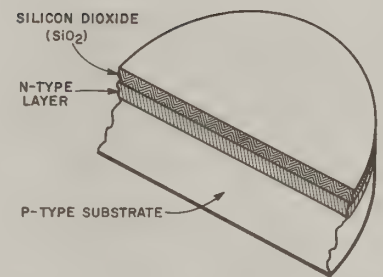
Monolithic IC's

The **MONOLITHIC** (or semiconductor) **IC** is a radical departure from the circuit board approach of the thin-film technology. It offers a truly "integrated" technology. All components (transistors, diodes, resistors, and capacitors) are formed within a single chip of semiconductor material, taking advantage of the fundamental electrical characteristics of materials and P-N junctions. A great number of diffusion, epitaxial, and other advanced fabrication techniques are used to manufacture monolithic IC's.

Industry is always striving for processes which will improve circuit operating characteristics, the number of components which can be placed on a single chip, and the economics of making and using IC's. The monolithic approach to IC construction may never achieve the wide range of component values and the close tolerances associated with thin-film circuits. The inherent size (Figure 4), reliability advantage, along with the low cost of incorporating great numbers of components in a single circuit, account for the predominant monolithic usage. This is especially true where great quantities of identical circuits, containing many active elements, are needed for

applications not requiring precision component values. The digital logic in computers is a primary example of such a requirement.

The active and passive components of the desired circuit are all formed within the N-type layer of the wafer structure shown in Figure 5. The thick P-type layer serves as a substrate material (holding everything together) and is also used to provide electrical isolation for the components in the N-layer. The silicon dioxide (SiO_2) layer is used to protect the components from corrosion, and also serves as an intercomponent insulating material. The N-type layer is very thin and may either be diffused into the P-type substrate or epitaxially grown onto this substrate. Epitaxial growth is a process whereby an existing single-crystal structure (the P-type substrate, in this case) is subjected to re-growth of the same or of a different type of substrate, to any desired molecular thickness. The process used is continued throughout the fabrication of all component regions.



BASIC MONOLITHIC IC WAFER STRUCTURE

Figure 5

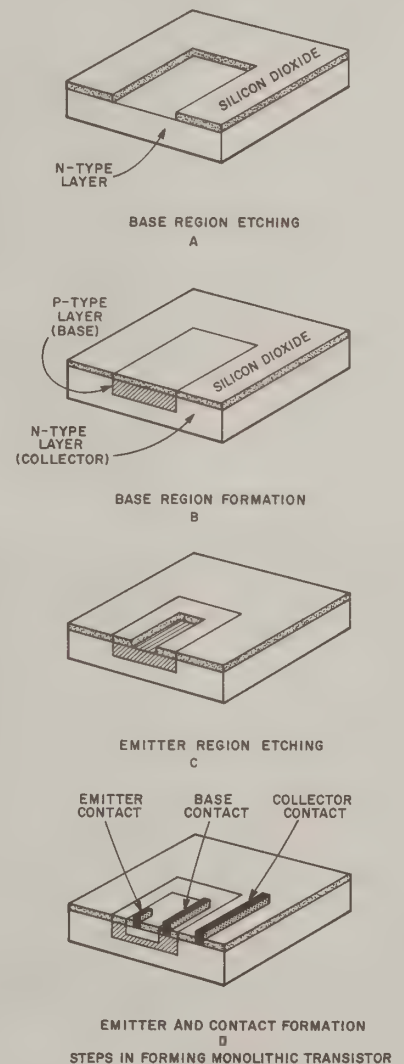


Figure 6

A basic approach to component construction, using either epitaxial growth or evaporation techniques, is shown in Figure 6. To form a transistor (the most complicated component), the original N-type layer serves as the collector region. The desired pattern for the transistor base region is etched through the silicon dioxide (Figure 6A). The vacant region is "filled" with a P-type material and again covered with silicon dioxide (Figure 6B). The desired pattern for the emitter region is etched through the new covering (Figure 6C). The vacant region is "filled" with an N-type material and again covered with silicon dioxide. As the last step, the various contact regions are etched through the silicon dioxide and a gold or aluminum contact is attached (Figure 6D). The formation of a diode only requires the first two steps (Figure 6A and B) and the attachment of contacts. A capacitor is formed in the same way, taking advantage of the P-N junction capacitance. Capacitor regions are generally larger than diode regions in order to achieve practical capacitance values. Resistors are formed by simply isolating a section of the N-layer from all other components and attaching conducting contacts. The length and width of the section is selected to achieve the desired resistance value. Multiple sections may be connected in series to obtain larger values. An alternative is to implant a P-type region of the desired length and width in the N-layer (Figure 6B) and attach contacts to this section. The P-N junction thus formed is reverse biased assuring that current will only flow through the P-type "resistor" and not through the junction.

The circuit in Figure 7 is simple, and contains far fewer components than can be fabricated on a single chip with present monolithic technology. It serves as a good example of how various portions of the different circuit components are formed (some simultaneously, some sequentially). In Figure 8A, the silicon dioxide is etched away in all regions except those which will contain the circuit components. The N-type regions, exposed by the etched portions, are changed to P-type regions down to a predetermined depth by a diffusion process, then they are recoated with silicon dioxide.

DIGITAL INTEGRATED CIRCUITS

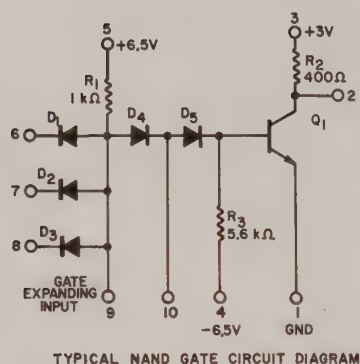
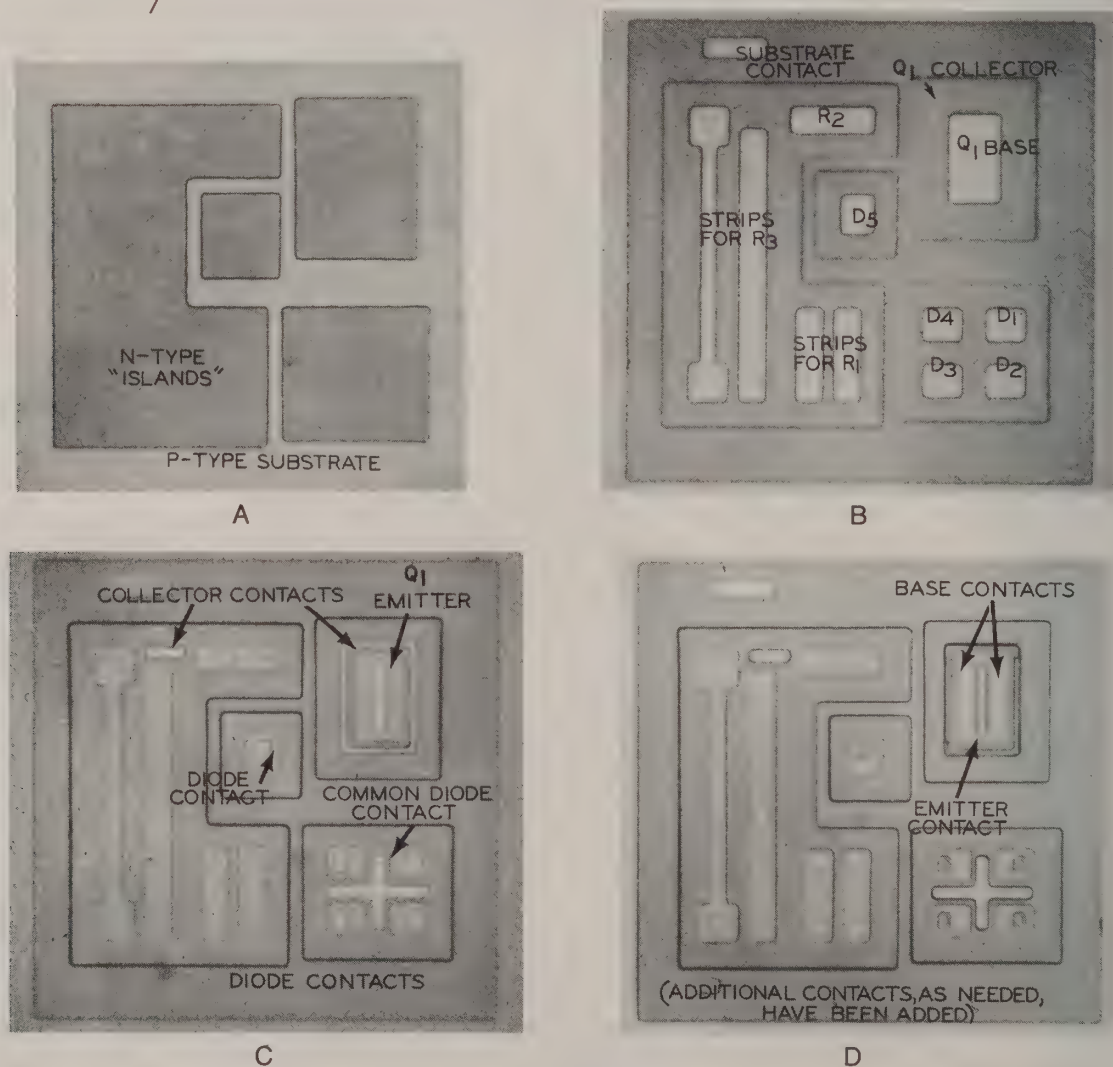


Figure
7

The entire P-type region is the most negatively biased portion of the circuit, thus assuring a reverse bias (and electrical isolation) between all of the N-type "islands" which have now been formed on the chip. Figure 8B shows the etching pattern for the P-type regions of all the components of the circuit. These regions are formed within the N-type islands by diffusing an appropriate dopant into the sections shown. All resistors and the base regions of the transistors are formed in this step. The surface is again covered with silicon dioxide to protect the new regions. If capacitors were included in the circuit, they would be formed in this step. In Figure 8C, openings are etched into the silicon dioxide for the transistor emitter region, and for all contact faces.

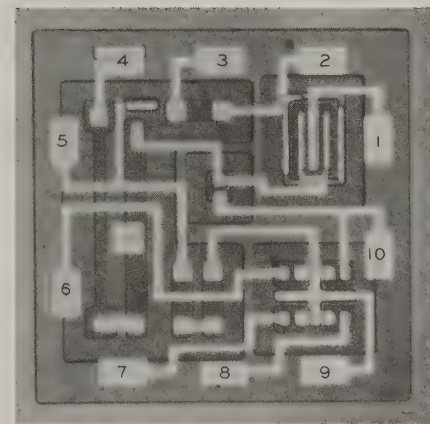


NAND Gate Fabrication on IC Chip
Courtesy Motorola Inc., Semiconductor Products Div.

Figure 8

The portion of the Q_1 base which is to become the emitter region is changed back to an N-type material (by using the diffusion process). Aluminum or gold film is used to make the contact points, resulting in the chip face shown in Figure 8D. (Notice how the common anode connection is made for D_1 through D_4 . Layout “tricks” of this type are an important part in the art of IC design, in order to obtain the most efficient usage of the chip face.)

The metal interconnection film has been added to complete the chip in Figure 9. Over most of its area the film lies atop a layer of silicon dioxide, and therefore is insulated from the underlying components. Electrical contact is made only at those points where contact faces are provided (Figure 8C). Notice that the two sections for R_1 and the two for R_3 are connected in series by a metal contact in order to obtain the desired $1\text{ k}\Omega$ and $5.6\text{ k}\Omega$ resistance values. The bonding pads (for external electrode connection) are numbered to correspond with the terminals of the original circuit diagram (Figure 9). The diffusion process is prevalent in current fabrication technology. Hundreds of circuits are formed, side-by-side, on a single semiconductor wafer about one inch in diameter. The chips are cut apart with a diamond scribe for mounting in individual encapsulations.



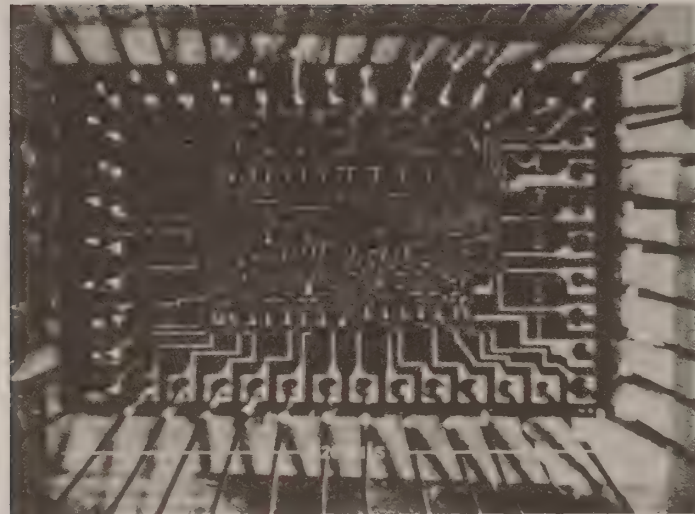
*Completed NAND Gate IC Chip
Courtesy Motorola Inc.,
Semiconductor Products Div.*

*Figure
9*

Since all of the components are formed from semiconductor materials, all characteristic values are sensitive to temperature and voltage variations. The passive components are designed within the limitations imposed by these materials. This and the desire for small size account for the limited range of resistance and capacitance values which can be achieved, as well as for the poor tolerances (about 20% for most circuits). Close tolerances can be achieved only at considerably increased cost. **PARASITIC CAPACITANCES**, existing every place in the chip where a P-type and N-type material lie in contact, produce high-frequency response limitations. However, new techniques overcoming these disadvantages are being continuously researched, perfected, and placed into manufacturing practice.

Medium- and Large-Scale Integration

It is difficult to imagine “cramming” more components onto an IC chip which already has as many as 2000 elements on its surface. (This represents the state of the monolithic IC capacity with conventional-type junction diodes and transistors.) A new technology, which is the basis for a “second generation” of IC devices, utilizes the metal-oxide-semiconductor field effect transistor (MOSFET). With this approach it is possible to obtain as many as 50,000 elements on a chip surface only slightly larger than the normal monolithic IC chip. The active devices (field effect transistors) furnish their own inherent interelement isolation; have a very high input impedance; and can be operated at high frequencies because they use an electric field for control, rather than a biased junction. An early MOSFET chip is shown with its bonded wires in Figure 10.



Early MOS Integrated Circuit
 Courtesy Charles E. Merrill Publishing Co.

Figure 10

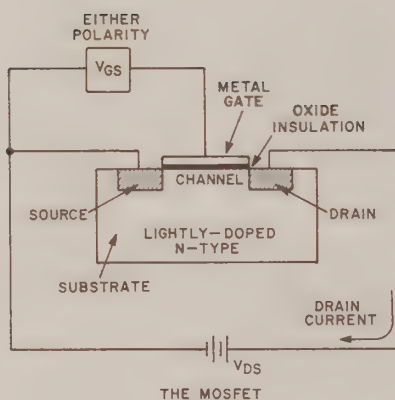


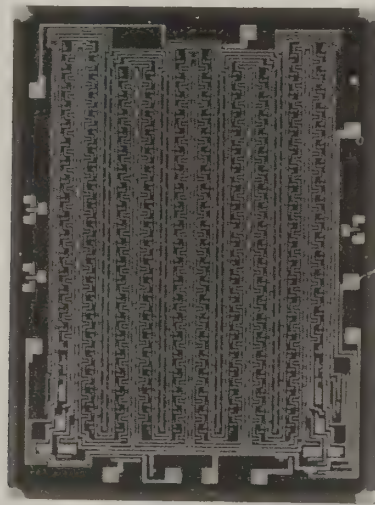
Figure
 11

The basic construction and circuit connection for a MOSFET is shown in Figure 11. A lightly-doped (almost intrinsic) silicon substrate, either N-type or P-type material, is used. This type of material has a light impurity concentration, and therefore has very high resistivity. During the fabrication process, two small regions are very heavily doped, forming a source and drain, similar in function to the emitter and collector regions of a conventional transistor. However, a junction is not formed since these regions are of the same type material as the substrate. A thin layer of silicon dioxide is deposited over the substrate, between the two heavily doped regions. A metal film is then deposited over the oxide, forming the gate element. The substrate region between the source and drain is called the channel and it serves as the transfer path for current carriers. Since the metal gate is not in direct electrical contact with the substrate or with the source and drain regions, no semiconductor junctions are formed.

In normal operation, there is current between the source and drain, through the channel region. The value of the current is dependent on the resistance of the channel region. The presence of a voltage between the metal film (gate) and the source controls the conduction through the channel region by setting up a capacitive effect and restricting the channel area available to the carriers. A large negative bias on the gate of an N-type or N-channel FET "cuts off" the device. This type of FET is known as a depletion FET. Therefore, a MOSFET may be used as a switching device (by placing alternate large negative and positive voltages on the gate). A P-channel FET functions similarly; however, voltages of opposite polarity are applied.

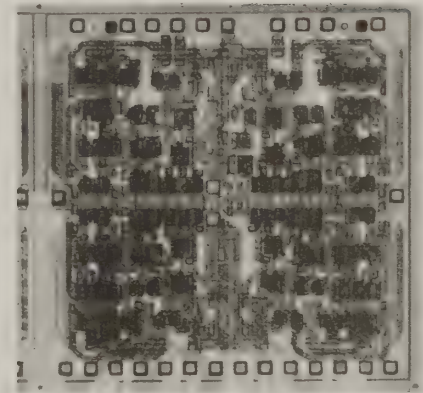
The MOSFET IC chip is “covered” with closely-spaced source, gate, and drain elements; and also with a metal interconnecting film. The interconnections are arranged to produce FET, resistive, and capacitive components by joining the proper elements. The interconnections also serve to join these components into the desired circuit network.

It is now possible to computer control chip layout. This results in the chip shown in Figure 12. Almost every “spot” on the chip is a separate element. With the computer approach, maximum utilization of the chip surface is achieved because of the regularity of the machine-generated patterns. There are so many possible devices on the chip that several layers of interconnecting film (each separated by an oxide-insulating layer) may be used in forming the complete circuit network (Figure 13). Visually, it is impossible to distinguish separate components and connections due to the many overlapping layers. This level of integration complexity is termed **LARGE-SCALE INTEGRATION (LSI)**, and when the component density is not as great, it is called **MEDIUM-SCALE INTEGRATION (MSI)**.



*Computer Designed MOS Chip
Courtesy Motorola Inc.,
Semiconductor Products Div.*

Figure 12



*Multi-layered MOS Chip
Courtesy Hayden Book Co.*

*Figure
13*

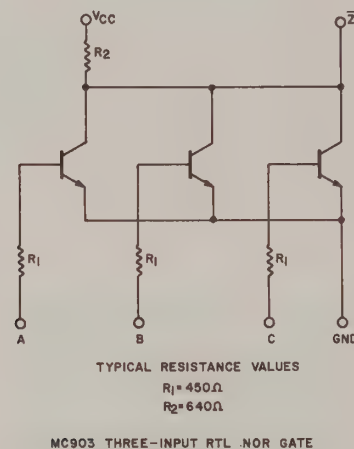
DIGITAL IC TYPES

Many types of logic circuitry have been developed with the advent of the integrated circuit. Each type has its advantages and disadvantages. Some of

the different logic types along with their advantages and disadvantages are discussed in this lesson.

Resistor-Transistor Logic

Parallel CE stages are used in the resistor-transistor logic (RTL) circuit shown in Figure 14. With a positive logic assignment, a NOR logic function would be produced.



Courtesy Motorola Inc., Semiconductor Products Div.

Figure 14

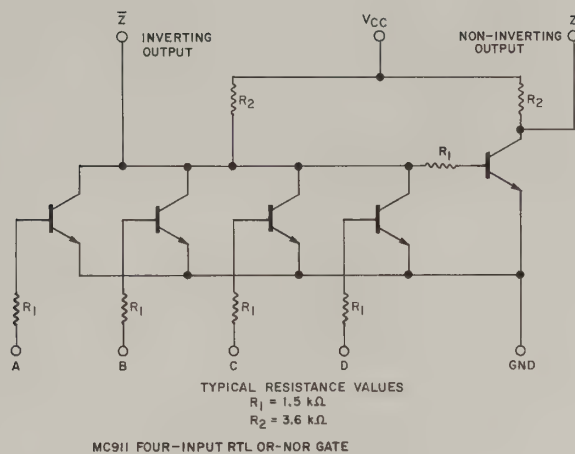
The input resistors used in RTL circuits range from 400 to 2000 ohms, making them larger than the transistor base resistance. Therefore, the base current is limited by the input resistor, not the transistor.

Figure 15 presents an RTL digital IC with two output terminals, thus enabling the circuit to supply both the OR and NOR outputs (with a positive logic assignment). The arrangements associated with the NOR terminal ($\bar{Z}_1$):

$$A + B + C + D = \bar{Z}$$

are identical to those in Figure 14, except for an additional parallel CE stage. For the OR terminal (Z_2), the NOR output signal is simply fed through an additional CE stage, resulting in double negation:

$$\overline{\overline{A + B + C + D}} = A + B + C + D = Z.$$



Courtesy Motorola Inc., Semiconductor Products Div.

Figure 15

AND and NAND gates are available (in limited numbers) in the RTL family of circuits. Accomplishing these functions with more than three inputs leads to very complex RTL circuit designs. This has considerably reduced the development of such circuits.

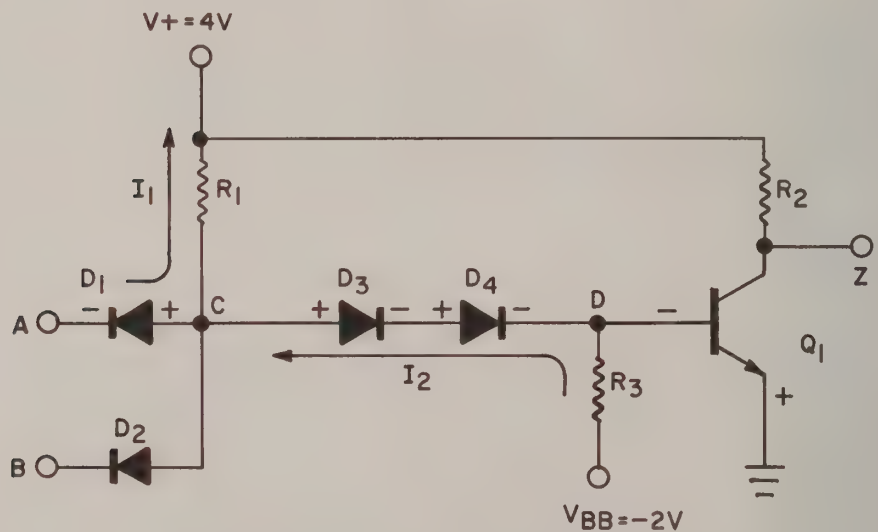
Diode-Transistor Logic

Prior to the development of integrated circuits, low and medium power RTL discrete circuits were the most popular form of logic. Because of its familiarity to logic designers, it was easiest to implement the RTL techniques into IC form when the new technology began expanding. In fact, most of the initial digital IC's were simply discrete RTL designs fabricated in IC form. Many of these are still available (along with other RTL designs developed specifically for IC's) even though other IC logic forms provide superior performance. Although the introduction of new RTL lines is extremely unlikely, the simplicity and low price of RTL IC's assures that they will continue to be available for some time to come.

After the initial development of digital IC's in RTL form, the use of diode-transistor logic (DTL) became quite popular. The use of diodes in digital IC's is a "natural" since any number of these devices can easily be formed along with the transistors when the chip is being fabricated. The use of input

diodes, rather than transistors, overcomes most of the problems associated with the RTL form, without introducing any major new problems. The "string" of diodes used to inject the signals provides a large input capability, and does not require separate transistor stages for each input condition. This, in conjunction with the use of both polarities of logic signals, makes a single DTL arrangement a very desirable and flexible system component.

Figure 16 illustrates a basic DTL gate circuit. With a low level, positive voltage applied to either or both of the input diodes (D_1 or D_2), the power supply side of the diode will be positive. This causes current I_1 to flow and a potential equal to one silicon "diode drop" (about 0.7 volt) appears at point C. This allows current I_2 from the base biasing source (V_{BB}), through diodes D_3 and D_4 , enabling the potential at point D to be two "diode drops" less than at point C (about -0.7 volt). With the base of the transistor being held at a slightly negative voltage (with respect to the emitter), the transistor is in its cutoff state, resulting in about +4 volts at the output.

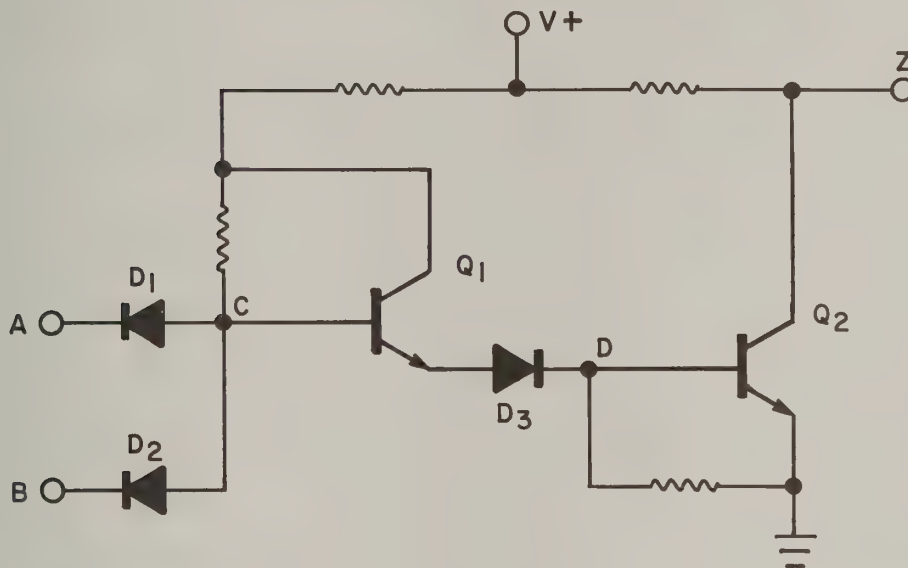


BASIC DTL GATE IN "OFF" CONDITION

Figure 16

When a positive voltage of 2 to 4 volts is applied to both input diodes, they become reverse biased and therefore stop conducting. This input signal level must be equal to or greater than +2.1 volts. Most of the source voltage is dropped across R_3 , enabling the base of the transistor to be at about +0.7 volt. The transistor switches into the saturation region, dropping the output

voltage to a low positive level (about 0.2 volt). Since this circuit will produce a logical 0 output only when both inputs are at logical 1, it is performing the NAND function (with positive logic).

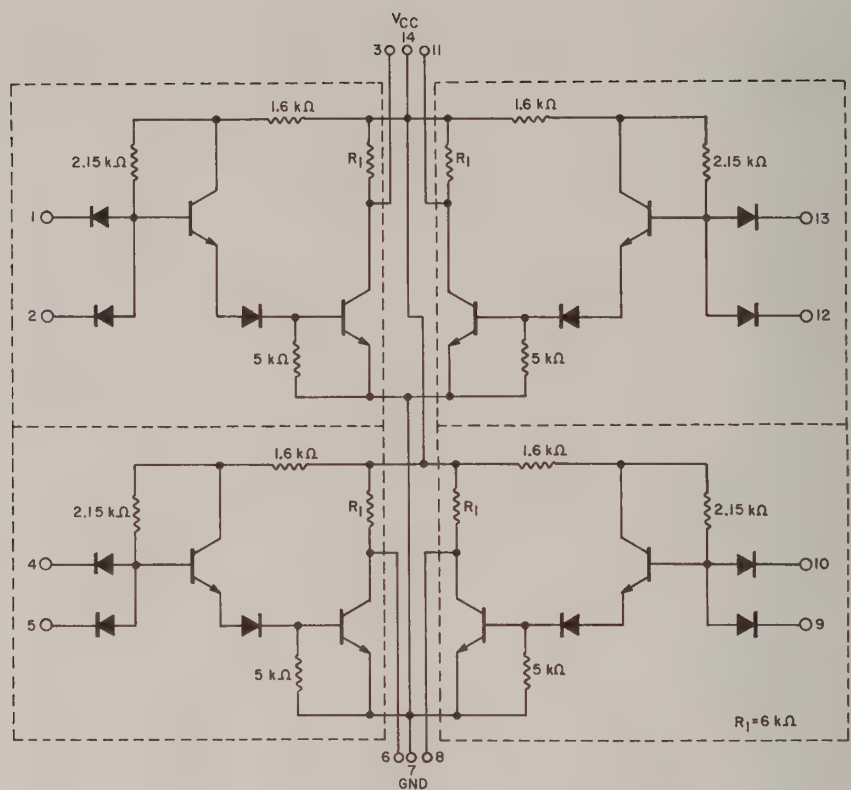


MODIFIED DTL CIRCUIT

Figure 17

The use of two power supplies in the basic DTL circuit has prevented this arrangement from being heavily utilized in digital IC's. The circuit can be arranged to operate with a single supply, and is available in IC form. However, certain penalties are incurred in terms of turn-off time, power dissipation, and fabrication cost. To avoid these penalties, the modified DTL circuit shown in Figure 17 has been developed. Diode D_3 of the basic circuit (Figure 16) has been replaced with a CC transistor stage, to serve as a driving element for the CE output stage. This fundamental arrangement forms the basis of most available DTL digital IC's, as shown within the dashed lines in Figure 18. This particular IC is a "quad" NAND gate with two inputs. A "quad" NAND gate has four identical NAND gates in the package. Here each gate has two inputs. Each package needs 14 pins: 8 inputs, 4 outputs, 1 ground, and 1 supply. It is not unusual for digital IC's to contain more than one identical gate on a single chip. Utilizing the terminal numbers (rather than letters) to indicate the input-output conditions, the circuit provides a positive logic NAND circuit as shown in the figure. With an inverted logic assignment, the circuits perform the NOR function. These conditions are also shown in the figure.

DIGITAL INTEGRATED CIRCUITS



A



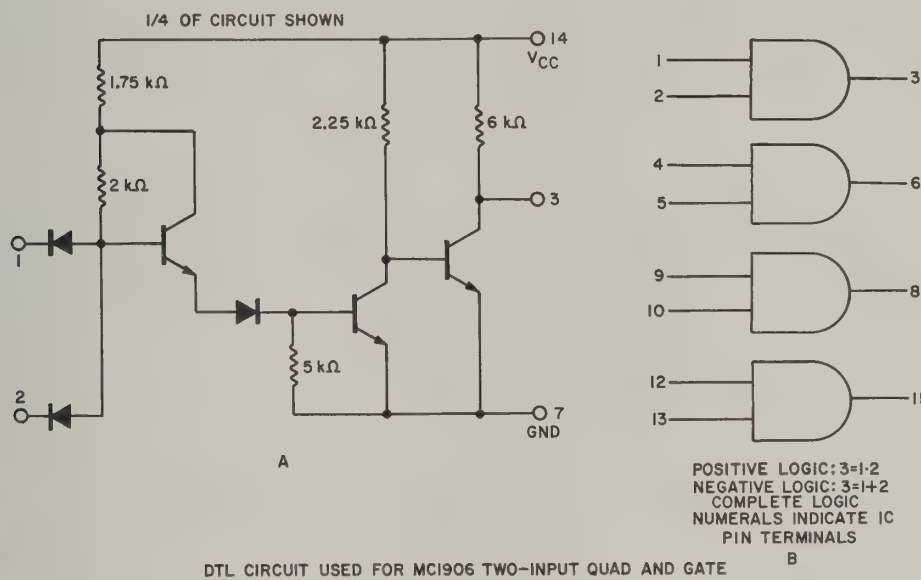
POSITIVE LOGIC: $3 = \overline{1 \cdot 2}$
 NEGATIVE LOGIC: $3 = 1 + 2$
 COMPLETE LOGIC
 NUMERALS INDICATE IC
 PIN TERMINALS

MC946 TWO-INPUT QUAD NAND GATE (DTL)

Courtesy Motorola Inc., Semiconductor Products Div.

Figure 18

Figure 19 shows the same fundamental circuit followed by an additional inverter stage. This inverter stage is used to form an AND gate with positive logic or an OR gate with negative logic. If the collector of the first inverter stage was connected to an output terminal, the circuit would perform both the NAND and AND operations (with positive logic) or both the NOR and OR operations (with negative logic).

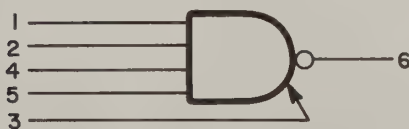


Courtesy Motorola Inc., Semiconductor Products Div.

Figure 19

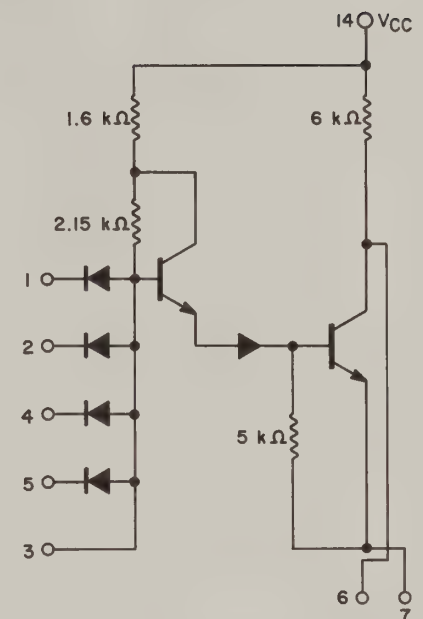
The digital IC shown in Figure 20 is a NAND gate with four inputs. The gate is an **EXPANDABLE GATE**; that is, each of the circuits is provided with an additional input terminal which can be fed by an external set of diodes, raising the number of input conditions for the gate. The symbol for an expandable gate indicates the additional terminal, as shown in Figure 21. The logic is written as:

$$1 \cdot 2 \cdot 4 \cdot 5 \cdot [3] = 6$$



SYMBOL FOR EXPANDABLE GATE

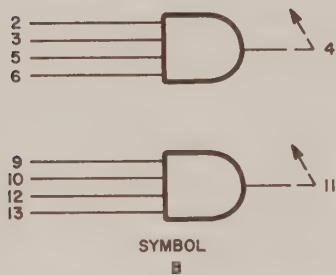
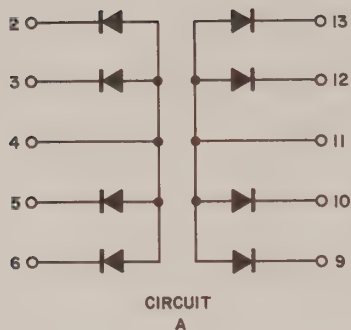
Figure 21



FOUR-INPUT NAND GATE (DTL) WITH EXPANDABLE INPUT

Courtesy Motorola Inc., Semiconductor Products Div.

Figure 20



DUAL FOUR-INPUT EXPANDER

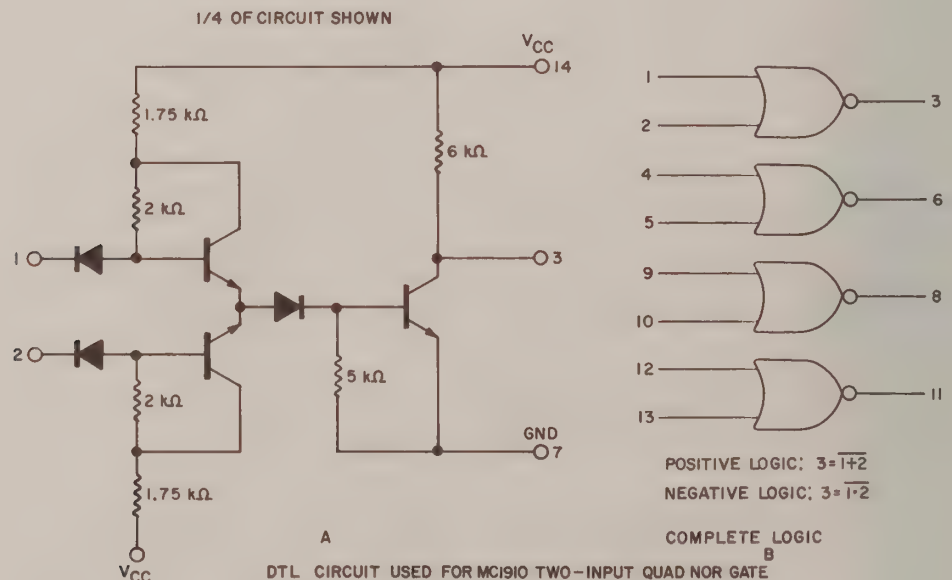
The bracketed terminal number indicates that this is not a normal input condition terminal. Figure 22 shows a dual gate expander, along with its logic symbol. If this is used with the dual expandable NAND gate in Figure 20, terminals 4 and 11 of the expander would be connected to terminal 3 of the NAND gate. The logic would then be:

$$1 \cdot 2 \cdot 4 \cdot 5 \cdot [2 \cdot 3 \cdot 5 \cdot 6] = 6$$

where the bracketed terminals refer to inputs applied at the expander. This has increased the number of inputs from four to eight.

Positive logic OR and NOR circuits are also available in the DTL form. It is not essential that one use inverted-logic AND and NAND circuits to obtain the dual functions. The basic NOR circuit is more complex, requiring separate CC stages for each input, as shown in Figure 23. This fundamental design, with one additional CE stage in the output, is used to obtain the OR function through double negation.

Figure
22



Courtesy Motorola Inc., Semiconductor Products Div.

Figure 23

The following Practice Exercise questions cover the subjects which you have just studied. They are:

INTEGRATED CIRCUITS

THIN-FILM IC'S

MONOLITHIC IC'S

MEDIUM- AND LARGE-SCALE INTEGRATION

DIGITAL IC TYPES

RESISTOR-TRANSISTOR LOGIC

DIODE-TRANSISTOR LOGIC

1. The NAND function, $\overline{A \cdot B \cdot C}$, is available from a particular circuit. How can this be converted to the AND function, $A \cdot B \cdot C$?
2. Why are the NOR and NAND functions preferred in the construction of digital IC's?
3. What purposes do the IC packages serve?
4. Why must the plastic encapsulated IC be operated in a less stringent environment than either the TO or flat metal or ceramic packages?
5. How is a capacitor formed in a monolithic IC?
6. What are parasitic capacitances?
7. MOSFET chips can be used to develop linear amplifiers, but cannot be used in digital switching applications. True or False?
8. To obtain a given logic function, such as OR, AND, NOR, etc., NPN and PNP transistor circuits must use logic assignments of _____ polarity.
9. How can one obtain the dual of the logical function of any given circuit?
10. A particular circuit produces the logic $\overline{A \cdot B \cdot C \cdot D} = Z$. How can the logic $\overline{A + B + C + D} = Z$ be obtained from this circuit?
11. Why are RTL IC's still available, although other forms offer superior performances?
12. The principle of duality does not apply to DTL. True or False?

13. What is meant by an “expandable” gate?
14. How can an IC manufacturer convert a given NOR or NAND circuit design into an OR or AND design, respectively?
15. Why are DTL OR/NOR circuits more complex than DTL AND/NAND circuits?

Transistor-Transistor Logic

Figure 24 shows the schematic for the input element (a multiemitter transistor) of **TRANSISTOR-TRANSISTOR LOGIC (TTL or T²L)** integrated circuits. This logic form provides high noise immunity, a large number of inputs and outputs, and high switching speeds. In effect, each of the emitter elements on this transistor acts as a separate emitter-base diode, all feeding into the same active amplifying device. This logic form is gaining extreme popularity in the field of digital IC's. The multiple emitters are easily formed during the fabrication process and take up far less space than the separate diodes of the DTL family. Since they are all formed simultaneously within a small region, they are usually very closely matched. In discrete circuit form, TTL would require a separate input transistor for each input condition.

In Figure 24 the supply voltage is often +5 volts, and the input levels are 0.2 volt and 3.3 volts. The potential at the base of Q_1 is determined by the lowest voltage at the input terminals. If A and B are at 3.3 volts and C is 0.2 volt, then the base of Q_1 is at or about 0.8 volt. This is because the emitter junction at C is forward biased while the two other emitter junctions are reverse biased. However, when terminal C is raised to the 3.3-volt level of the other two terminals, then the base of Q_1 is at about 2 volts. Whenever all three terminals are at 0.2 volt, the Q_1 base will be around 0.8 volt.

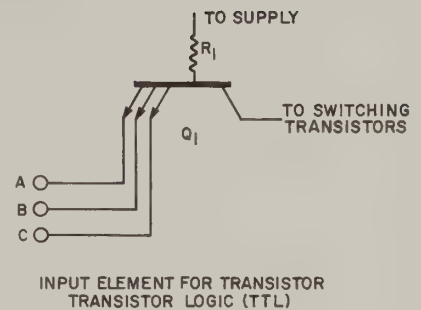
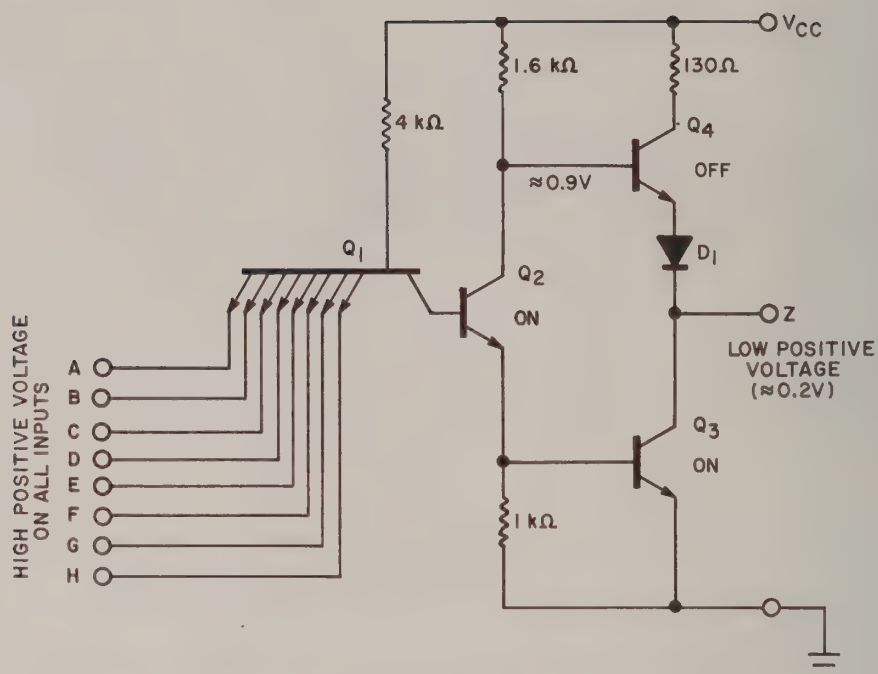


Figure
24

Figure 25 illustrates one of the most common basic forms of TTL. Although not shown, the proper loads are considered to be connected to the output. With a high positive voltage (typically 3.3 volts) on all of the gate inputs, Q_1 base is at about 2 volts and Q_2 and Q_3 are conducting at their saturation level, enabling the output to be at a low positive voltage (about 0.2 volt). Q_4 is cut off.

This causes the collector of Q_2 (and the base of Q_4) to be at a level just under one volt ($V_{BE(Sat)}$ on $Q_3 + V_{CE(Sat)}$ on $Q_2 = 0.9$ volt). With $V_{CE(Sat)}$ (about 0.2 volt) on the collector of Q_3 , the base of Q_4 is about 0.7 volt positive with respect to the collector of Q_3 . It takes about 1.4 volts between these two points ($V_{BD(Sat)}$ on $Q_4 + V$ on $D_1 = 1.4$ volts) to turn Q_4 on.

If the voltage on any one (or any combination) of the input emitters falls to a low positive value (typically 0.8 volt or less), the collector junction of Q_1 becomes reverse biased, and Q_2 and Q_3 go into cutoff. As Q_2 turns off, the base voltage of Q_4 rises sharply while the collector voltage of Q_3 is held at 0.2 volt for a very short time due to circuit capacitances. As a result, Q_4



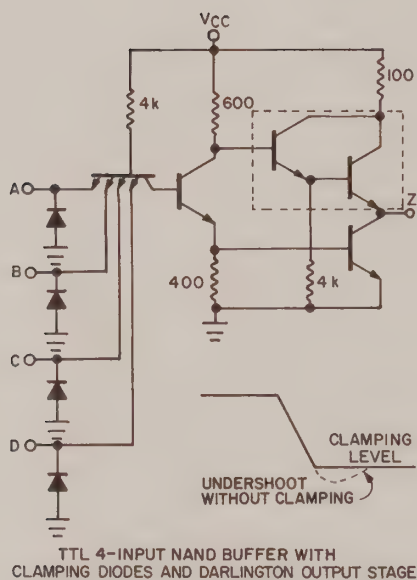
SN5430 EIGHT-INPUT NAND GATE (TTL) IN LOW OUTPUT CONDITION

Courtesy Texas Instruments

Figure 25

rapidly goes into saturation, conducting through the loads, allowing an extremely rapid buildup of the output voltage (through the 130 Ω resistor and the slight resistance of Q_4 and D_1). The reverse procedure occurs when all inputs switch to logic 1. This approach circumvents the storage time associated with Q_2 and Q_3 , which accounts for most of the turn-off delay in other types of logic circuits.

Many TTL circuits utilize additional diodes in the input portion of the circuit, as shown in Figure 26. These diodes provide a clamping action which prevents undershoot during the turn-off operation (also shown in the figure). This phenomenon, also known as "ringing," becomes especially bothersome in applications requiring long interconnective wiring.



Courtesy National Semiconductor Corp.

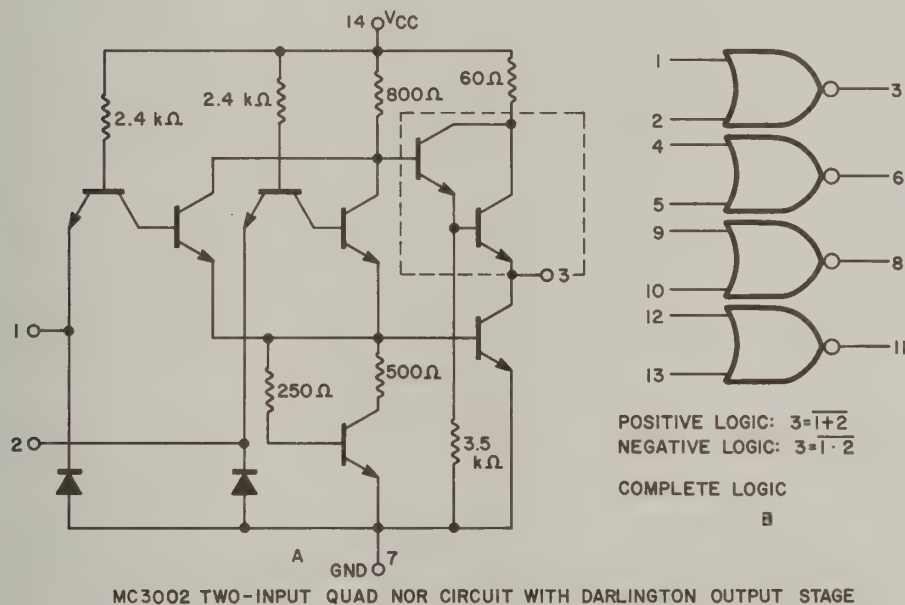
Figure 26

The circuit shown in Figure 26 is called a **BUFFER**. It is designed to operate with three times as many circuits connected to its output as ordinary TTL NAND gates. The circuit shown in Figure 26 also contains an additional output transistor (in the dashed box) forming a **DARLINGTON CONFIGURATION**. The Darlington configuration is one of the more recent innovations in TTL which is being used for most new TTL circuits, regardless of their function. The Darlington configuration provides a high input

impedance for the input "driving" transistors, and a very low output impedance.

The NAND gate shown in Figure 27 is often used when saturated circuitry with high switching speeds is required. In order to achieve the high switching speeds, the resistors are of lower value than those in the standard lower switching speed TTL circuits. Consequently, the high speed TTL gates dissipate more power than the standard TTL gates.

TTL positive logic NOR and OR gates require a rearrangement of the input terminals through separate transistors, as shown in Figure 28A. The logic diagram is shown in Figure 28B; there are four separate NOR gates in the package.

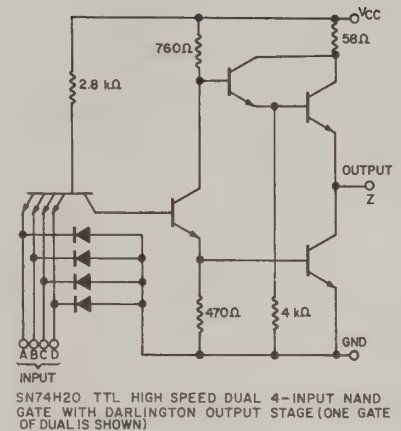


Courtesy Motorola Inc., Semiconductor Products Div.

Figure 28

The gate of Figure 28 can be easily changed into a two-level logic circuit by adding two more input diodes, one more emitter junction for each input transistor, and two more input terminals. The resultant circuit is shown in Figure 29A, and the logic diagram is shown in Figure 29B.

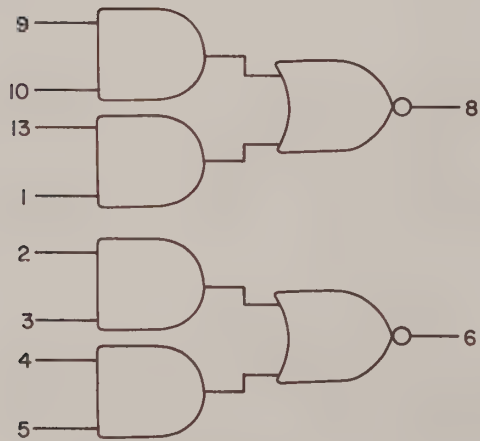
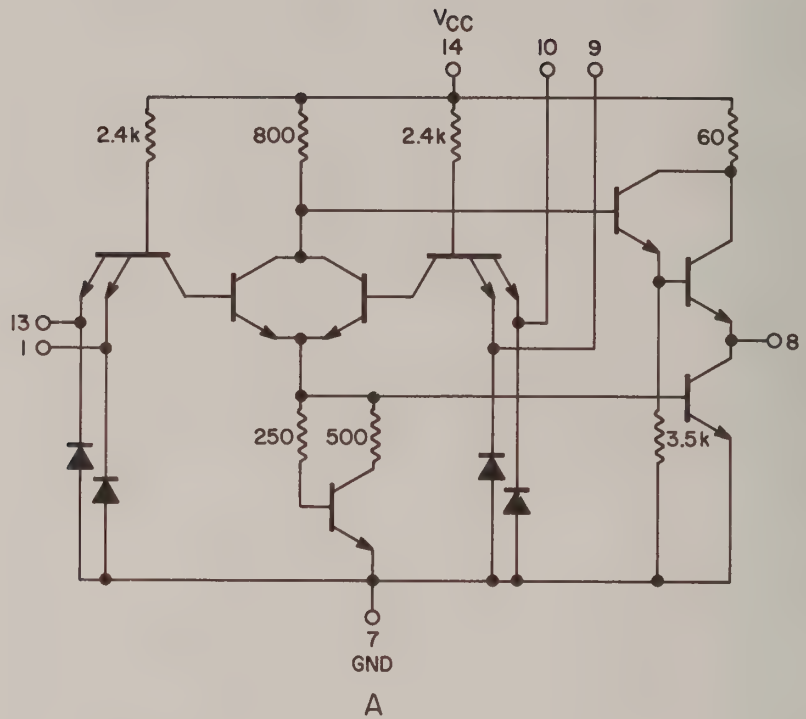
The circuit shown in Figure 30 is known as “super high speed,” or Schottky, TTL. It utilizes Schottky-barrier diode clamping on all normally saturated transistors. The diodes prevent the transistors from operating in saturation;



Courtesy Texas Instruments

Figure 27

DIGITAL INTEGRATED CIRCUITS



POSITIVE LOGIC:

$$8 = (9 \cdot 10) + (13 \cdot 1)$$

NEGATIVE LOGIC:

$$8 = (9 + 10) \cdot (13 + 1)$$

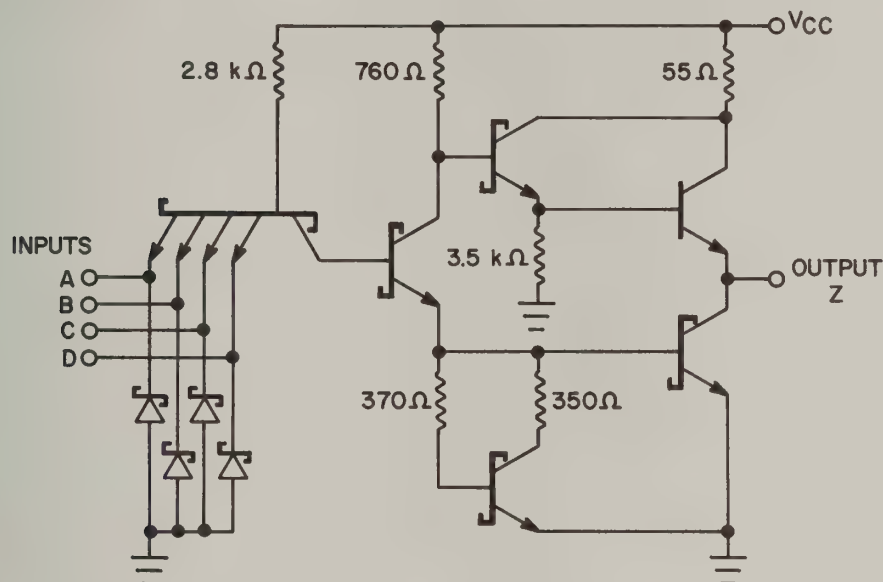
B

MC3023 DUAL 2-WIDE 2-INPUT "AND-OR-INVERT"
TTL HIGH SPEED GATE

Courtesy Motorola Inc., Semiconductor Products Div.

Figure 29

therefore, Schottky TTL is a form of unsaturated TTL. Because of the reduction of the storage time, the delays encountered by Schottky TTL are about one half of high speed TTL, and one third of the standard TTL. Schottky circuit dissipation is the same as high speed TTL circuit dissipation. The special symbols shown in Figure 30 represent the Schottky high-speed transistors and diodes.



74520 TTL SUPER HIGH SPEED (SCHOTTKY) DUAL 4-INPUT NAND GATE

Courtesy Fairchild Semiconductor

Figure 30

The unused inputs of any TTL circuit should not be left "open;" therefore, they should either be tied to the used inputs or returned to the supply voltage. This reduces the possibility of external noise signals being coupled into the circuit. If the inputs are returned to the supply, the supply voltage must not fluctuate above the maximum input voltage rating. If tied to the used inputs, the "unused" inputs act as additional loads on the driving element.

Nonsaturated Logic

The complete elimination of the storage time effect in switching can only be obtained by not driving the transistors into the saturation mode. A form of nonsaturated logic is obtained with switching circuits termed current mode logic (CML), or emitter-coupled logic (ECL). These circuits are designed so that the high and low voltage states are standard fixed values other than near-zero and $V_{CE(Sat)}$. Therefore, the voltage swing is smaller, and the switching

speed is considerably enhanced. CML's biggest detriment is a low noise immunity (making it more sensitive to erroneous switching than any of the other logic families) and multiple-supply requirements. The use of a large number of transistors in these circuits has prevented CML from being popular in discrete form. However in digital IC's, it has received considerable attention.

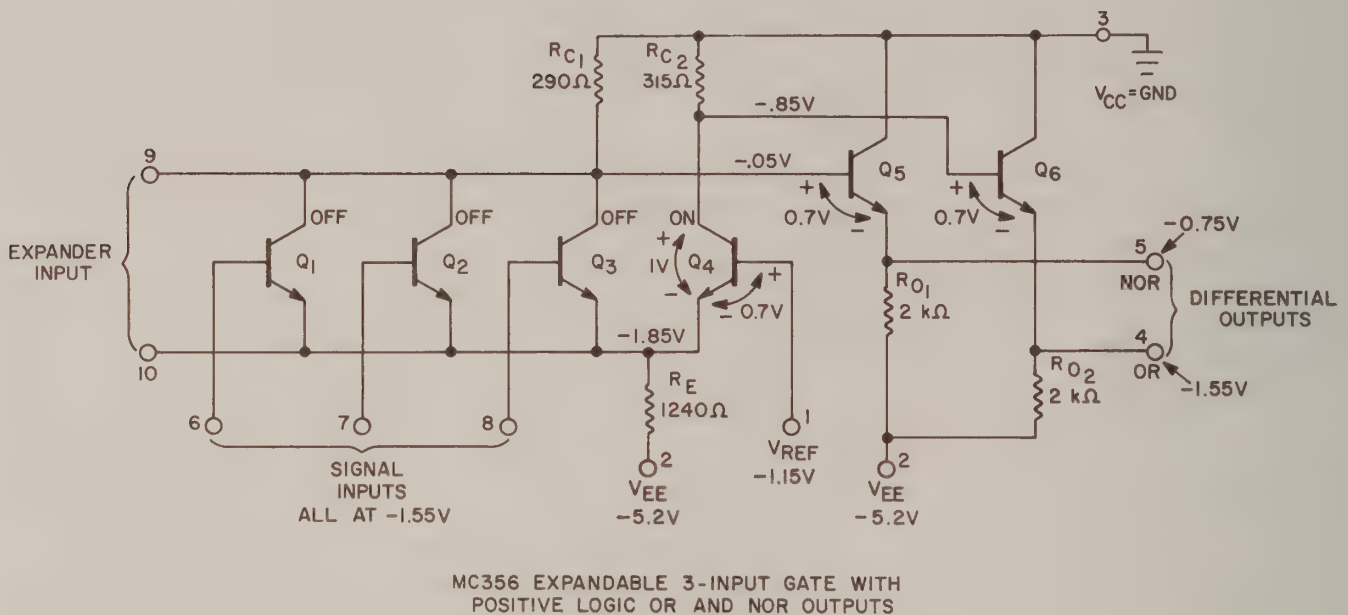


Figure 31

A typical nonsaturated digital IC circuit is shown in Figure 31. It consists of a **DIFFERENTIAL AMPLIFIER** made up of a three-input phase splitter and a CC output section. The three signal input transistors, Q_1 , Q_2 , and Q_3 in the differential amplifier, are reverse biased and are in the "off" condition. Q_4 is forward biased and is in the "on" condition. With NPN transistors, this is basically a positive logic circuit, since the low negative voltage level is a logical 1 and the high negative voltage level is logical 0. The single circuit provides both OR and NOR outputs. With inverted logic assignments, AND and NAND outputs can be obtained from the same circuit.

Since Q_4 is conducting and its base is at -1.15 volts, its emitter will be 0.7 volt more negative or -1.85 volts. Consequently, there is $5.2 - 1.85$ or 3.35 volts across R_E . The current through R_E is 2.7 mA. Since Q_1 , Q_2 , and Q_3 are cut off, the current through them is very small so that the voltage drop across R_{C1} is about 0.05 volt. We can say that all of the current through R_E also flows through Q_4 and R_{C2} . There is 315 ohms times 2.7 mA, or 0.85 volt, across R_{C2} . The OR terminal voltage is V_{BE} below -0.85 volt (or -1.55 volts). The NOR output is V_{BE} below -0.05 volt (or -0.75 volt). The drop across R_{C2} and R_E is $0.85 + 3.35$ or 4.20 volts. Therefore, V_{CE} of Q_4 is $5.2 -$

4.2 or 1 volt. Q_4 is now operating in the active region instead of the saturation region.

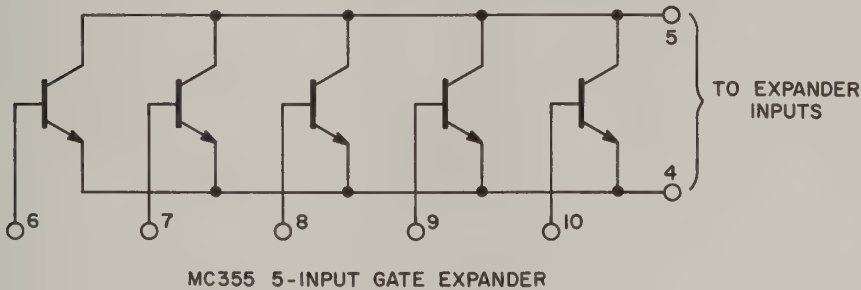


Figure 32

The circuit shown in Figure 32 is called a gate expander. When used with the circuit shown in Figure 31, a total of eight inputs can be obtained. Pins 5 and 4 of the gate expander would be connected to pins 9 and 10, respectively, of the expandable three-input gate circuit shown in Figure 31. One or two more gate expanders may be added to the gate circuit in Figure 31 to provide a total of 18 inputs. The gate expander outputs would then be paralleled together (all pins 5 connected together and all pins 4 connected together) before connecting the gate expander circuit to the input pins (9 and 10 respectively) of the gate circuit.

Since a differential amplifier is used in this ECL circuit, a reference voltage is required. The reference voltage is applied to pin 1 of the gate circuit shown in Figure 31. If a regulated dc potential of -1.15 volts is not available, the IC bias driver shown in Figure 33 can be used. Pin 1 of the bias driver would then be connected to pin 1 of the gate circuit.

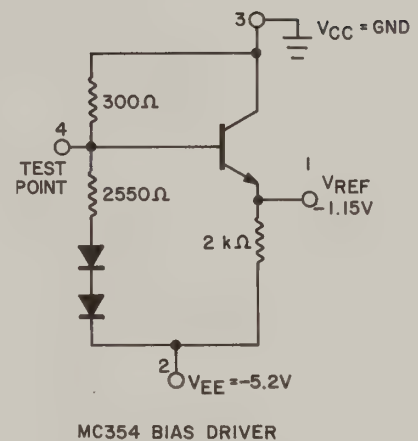


Figure 33

By applying -0.75 volt to the base of Q_3 in Figure 31, the emitter junction of Q_3 becomes forward biased. The emitters of Q_1 , Q_2 , and Q_4 are reverse biased. The base of Q_3 will be -0.75 volt, while the emitter is 0.7 volt more negative, or -1.45 volts. The voltage drop across R_E becomes 5.2 volts -1.45 volts, or 3.75 volts. The current through R_E is approximately 3 mA. There is no current through R_{C2} , since Q_4 is cut off. Therefore, 3 mA flows through R_{C1} , developing approximately 0.85 volt. The NOR output is V_{BE} below 0.85 volt, or -1.55 volts. The OR output is V_{BE} below 0.05 volt or -0.75 volt. The voltage drop across Q_3 is 0.6 volt; therefore, it is operating in its active region. Q_4 will be cut off when any or all of the input transistors (Q_1 , Q_2 , and Q_3) have a logical 1 (-0.75 volt) applied to their inputs.

The unused inputs of ECL gates should be connected to V_{EE} in order to avoid the introduction of external noise signals. The high input resistance and very low output resistance makes ECL especially adaptable to large input and output requirements.

Metal Oxide Semiconductor (MOS) and Complementary Metal Oxide Semiconductor (CMOS) Logic

The basic mode of operation of field effect transistors in MOS logic circuits is the voltage switch (or "inverter logic"). The circuit of a typical MOS inverter is shown in Figure 34A. The P-channel MOSFET is connected to the negative supply voltage through a "pull-up resistor." Figure 34B shows that the pull-up resistor can be replaced by a special MOSFET current generator. This special MOSFET is electrically and economically superior to the resistor. A potential of approximately -12 or -13 volts is used for $-V_{DD}$. In Figure 34B the gate of the top MOSFET can be connected to $-V_{DD}$ or to a higher negative potential of approximately -25 or -30 volts. Figure 34C shows the waveforms of either inverter circuit when negative logic is used. The two inverters draw current only when a negative voltage is applied to the gate of the input MOSFET. Enhancement mode MOSFET's are very compatible for this inverter switch because there are few problems with voltage levels and impedances. Figure 34D shows the transfer curves associated with the inverter circuits. A typical value for logic 0 is -0.8 or -1 volt; and for logic 1 is -12 volts. Two simple gates are illustrated in Figure

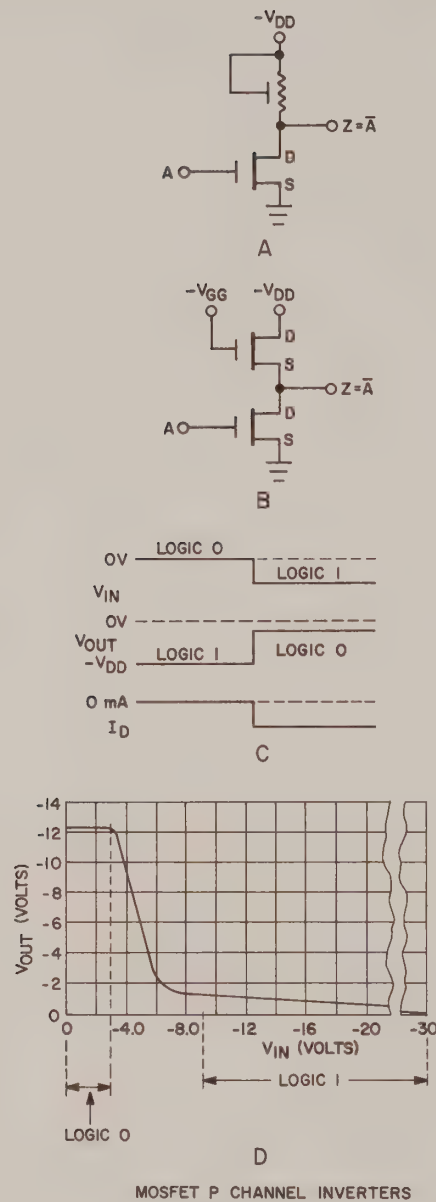


Figure 34

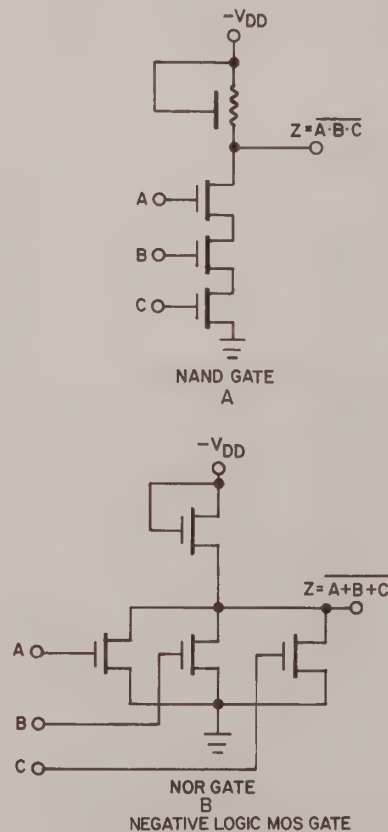


Figure 35

A	B	C	Z
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	1	1
1	1	0	1
1	0	0	1
1	1	1	0

Truth Table for
Circuit in Figure 35A

Table 1

35. In Figure 35A a special MOSFET can also be used in place of the "pull-up resistor." Table 1 is the truth table for the NAND circuit of Figure 35A, and Table 2 represents the circuit of Figure 35B.

Eliminating one input transistor of the gate circuit shown in Figure 35A, replacing the resistor by a MOSFET and adding two MOSFET's connected in series and placed in parallel with the first two MOSFET's, results in the two-level gate circuit shown in Figure 36A. The logic diagram of the two-level gate circuit is shown in Figure 36B.

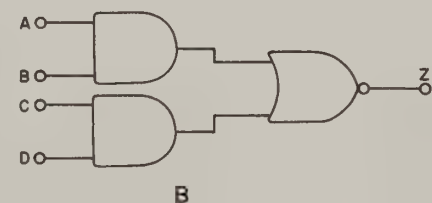
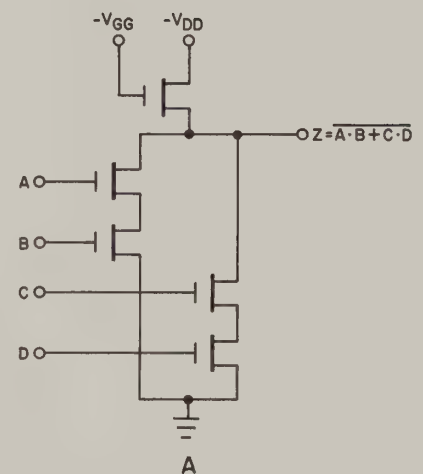
The series and parallel MOSFET circuit combinations shown in Figure 35 can be connected together to form an almost unlimited number of different circuits. Some examples of these circuits are: EXCLUSIVE-OR, half and full adders and subtractors, buffers, multi-phase inverters, and many types of memory circuits including static and dynamic shift registers, all types of flip-flops and latches.

In the digital circuits shown in Figures 34, 35, and 36 P channel MOSFET's are employed. N channel MOSFET's can also be used by reversing the supply voltages. The simplest CMOS circuit is the inverter shown

A	B	C	Z
1	1	1	0
1	1	0	0
1	0	1	0
1	0	0	0
0	1	0	0
0	0	1	0
0	1	1	0
0	0	0	1

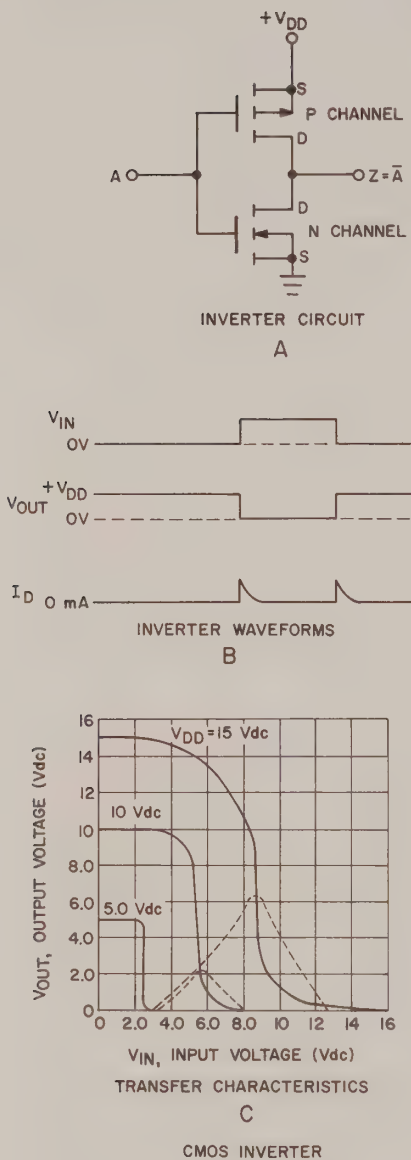
Truth Table for
Circuit in Figure 35B

Table 2

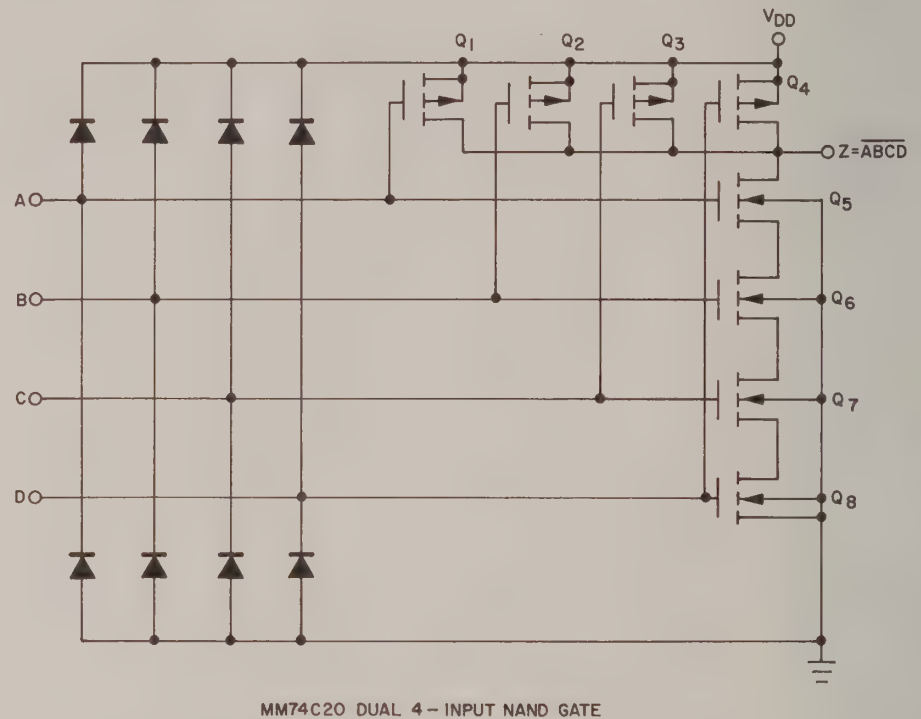


TWO LEVEL MOS GATE

Figure
36



in Figure 37A. Circuit operation is quite simple: when the voltage at A is near zero volts, the P-channel FET is on while the N-channel FET is off and the output is near V_{DD} . However, when the input voltage is near V_{DD} , the two FET's change states, causing the output to drop to near zero volts. The circuit current is very low in the nano-ampere range, except for very brief intervals when switching occurs. The inverter waveforms are shown in Figure 37B. The supply voltage can be between 3 and 15 volts. The transfer curves for the inverter with three different supply voltages are shown in Figure 37C.



Courtesy National Semiconductor Corp.

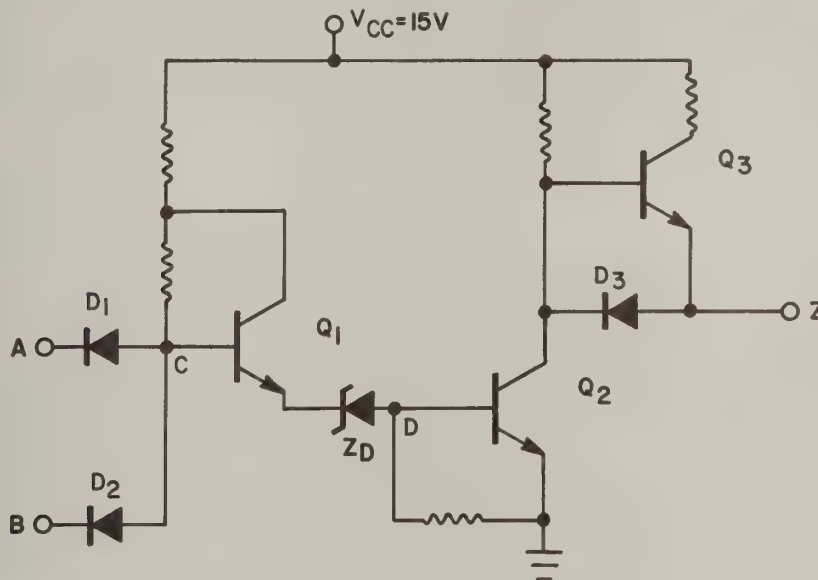
Figure 38

Figure 37

A dual 4-input NAND gate utilizing CMOS logic is shown in Figure 38. The operation of this gate circuit is similar to that of the inverter. When all four inputs are high (near V_{DD}), Q_1 through Q_4 (P-channel FET's) are cut off and the other four devices (N-channel FET's) are conducting. Thus, the output voltage is near zero volts. If the voltage level at input B were switched to zero volts, Q_6 would become cut off, and Q_2 would conduct. The output would then have to switch from zero volts to near V_{DD} . If any other inputs were switched to zero volts, more P-channel MOSFET's would conduct and more N-channel MOSFET's would go into cutoff. However, the output would remain near V_{DD} . This operation meets the definition of positive logic NAND gates.

Other IC Logic Types

When extremely high noise immunity is required without high-speed operation, saturated switching circuits called **HIGH THRESHOLD LOGIC (HTL, HDTL, or HTTL)** can be used. This logic family utilizes zener diodes to raise the switching thresholds. A basic HDTL circuit is shown in Figure 39. In comparison to the DTL circuit (Figure 17), the diode between Q_1 and Q_2 is simply replaced with a zener diode. Now transistor Q_2 cannot conduct until the signal exceeds the normal DTL requirement by the zener voltage. A variety of zener voltages are available, but the most commonly used is 6.8 volts. This raises the high-state noise margin considerably. In order to raise the low-state noise margin, a high (usually 15 volts) supply voltage is used. Switching times are on the order of a few tenths of a microsecond; therefore, HTL can only provide a very low-speed switching capability.



HIGH THRESHOLD DIODE TRANSISTOR LOGIC CIRCUIT

Figure 39

Current mode logic (CML), also referred to as current steering logic (CSL), is shown in Figure 40. This type of circuit uses one transistor (Q_1 to Q_4) for every input (A to D) and an additional transistor (Q_5) for biasing. The circuit is designed so that current flows either through the bank of input transistors or through the biasing transistor. Thus, both the inverted and noninverted outputs are always present. Since R_3 limits the circuit current, complete saturation is not obtained, resulting in faster circuit operation than for fully saturated gates. The circuit of Figure 40 is both OR and NOR. If one or more inputs go positive, the OR output goes positive and the NOR output goes negative.

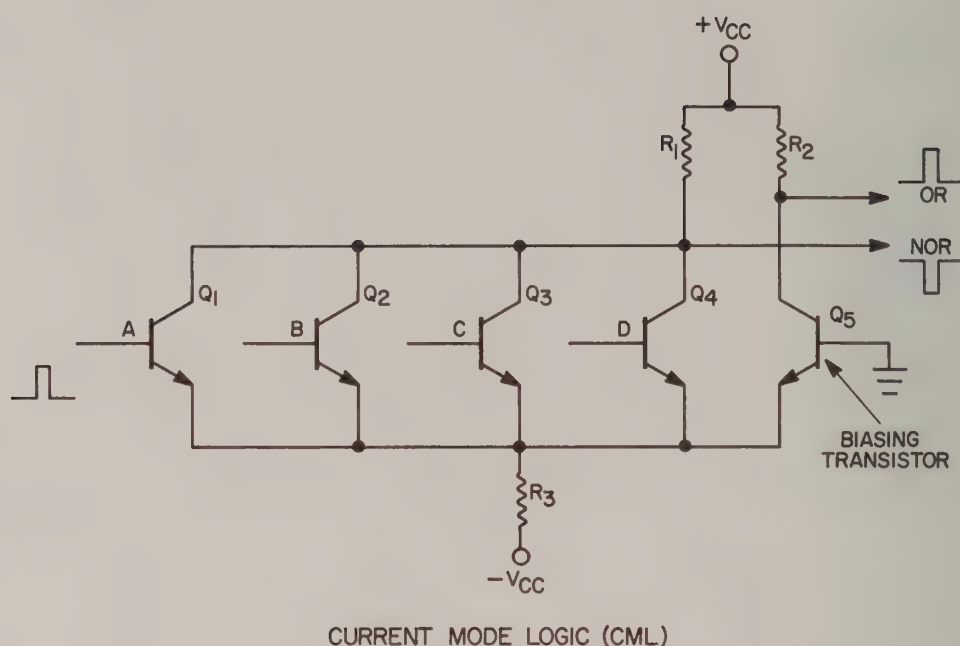


Figure 40

OPERATING CHARACTERISTICS

Every IC manufacturer provides a slightly different format and degree of detail to describe the performance and use of his particular IC's. General characteristics usually will be itemized in tabular form on a standard data sheet. The more general characteristics include power dissipation, supply voltage operating range, output driving capability, output impedance, output voltage swing, input voltage limits, switching threshold, input impedance, noise margin, switching speeds, input loading factor, and thermal limits and characteristics.

Temperature and Dissipation

As with any electronic circuit, a maximum allowable current and power dissipation will be specified for each IC. Current limits may range from less than 100 μ amps to a few hundred milliamps; power dissipation limits, from about 100 μ watts to several hundred milliwatts—but usually less than 1 watt. Allowable power limits are specified at room temperature. A derating factor, specified in milliwatts/ $^{\circ}$ C, must be applied when higher ambient temperatures will be experienced. This will degrade the allowable power limit by a specific amount per degree excess temperature. For example, a device with a 50 mW capability at room temperature (25° C) and a 2 mW/ $^{\circ}$ C de-

rating factor from 25°C can only handle 30 mW at an ambient temperature of 35°C:

$$\begin{aligned} P_{\max} &= 50 \text{ mW} - (2 \text{ mW/}^{\circ}\text{C}) (35^{\circ}\text{C} - 25^{\circ}\text{C}) \\ &= 50 \text{ mW} - 20 \text{ mW} = 30 \text{ mW.} \end{aligned}$$

Certain IC logic circuits (especially TTL and ECL) have both dc and ac specifications. The total dissipation for such devices is the sum of these two powers.

Since IC's are semiconductor devices, temperature limits are very important to their operation. This is why an operating temperature range should always be determined before using a particular IC logic circuit. This range will be specified so the junctions in the device will not exceed a predetermined maximum junction temperature.

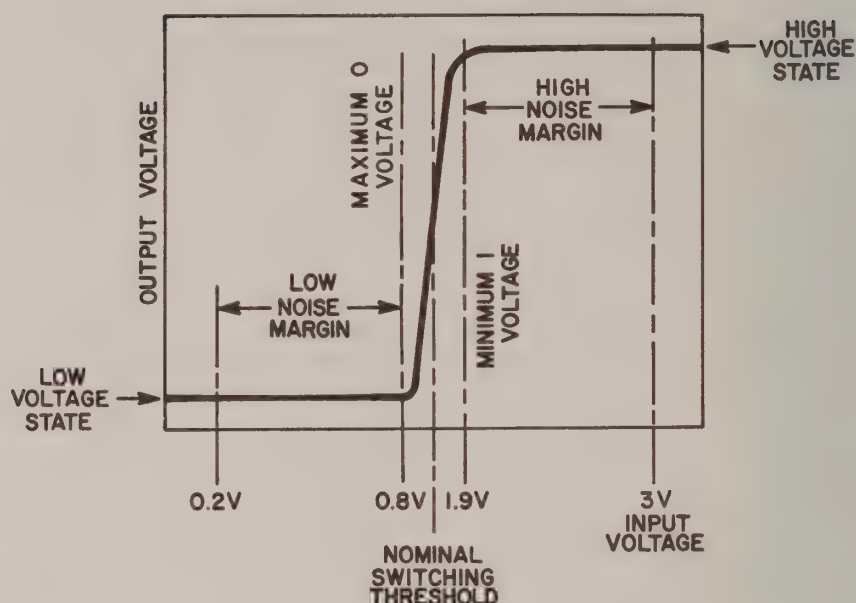
Voltage Swing and Noise

A particular logic system will have a defined set of voltage levels normally used to represent logical 1 and logical 0. For example, in a given type of circuit, the logical 1 may normally be defined by a level of +3 volts; the logical 0 by a level of +0.2 volt (positive logic). These levels are determined during the circuit design, and depend on the supply voltage and the transistor switching characteristics. The relative voltage levels are simply termed the "high" and "low" states. The actual input voltage values which cause a transistor to switch from one state to another are called the **SWITCHING THRESHOLDS**.

These are usually specified as the **MAXIMUM 0 INPUT VOLTAGE** and the **MINIMUM 1 INPUT VOLTAGE**. For a circuit with the logic levels described in the preceding paragraph, the maximum 0 input voltage might be +0.8 volt; the minimum 1 input voltage might be +1.9 volts. The input signal must go from +0.2 to +0.8 volt before logical 0 becomes logical 1, and must fall from +3 to +1.9 volts before logical 1 becomes logical 0. These non-inverting logic characteristics are all shown in Figure 41.

The difference between the normal logic level voltages and the threshold levels defines the circuit **NOISE MARGINS**. A logical 1 noise margin is defined by the following:

$$1 \text{ noise margin} = \text{normal 1 voltage} - \text{minimum 1 voltage.} \quad (3)$$



SPECIFIC POINTS DEFINED ON TRANSFER CHARACTERISTIC

Figure 41

For the example values used in the preceding paragraphs, the 1 noise margin is $3\text{ V} - 1.9\text{ V} = 1.1$ volts. There is also a logical 0 noise margin which is defined by the following:

$$0 \text{ noise margin} = \text{maximum 0 voltage} - \text{normal 0 voltage.} \quad (4)$$

For the example values used in the preceding paragraphs, the 0 noise margin is $0.8\text{ V} - 0.2\text{ V} = 0.6$ volt. Since the noise in a circuit may cause unwanted switching actions, it is desirable for digital IC's to have a high noise immunity. The smaller of the two noise margins indicates the noise immunity of a circuit because it represents the peak amplitude that the noise signal must reach in order to produce switching errors. For a circuit with the noise margins just described, immunity is 0.6 volt.

Fan-In and Fan-Out

FAN-IN is defined as the number of possible independent inputs to a logic unit established by the circuit design. Gate expanders are available and can be used in conjunction with OR, AND, NOR, and NAND circuits in order to increase the basic circuit fan-in. When this is done, each additional input increases the minimum 1 input voltage requirement by a specific amount. Each additional input also decreases the allowable maximum 0 input voltage

by a specific amount. Within limits, the increased input voltage requirements can be met by increasing the output voltage of the driving units (those circuits supplying the input signals). The decreased maximum 0 input voltage lowers the 0 noise margin for the circuit, making it more susceptible to erroneous switching.

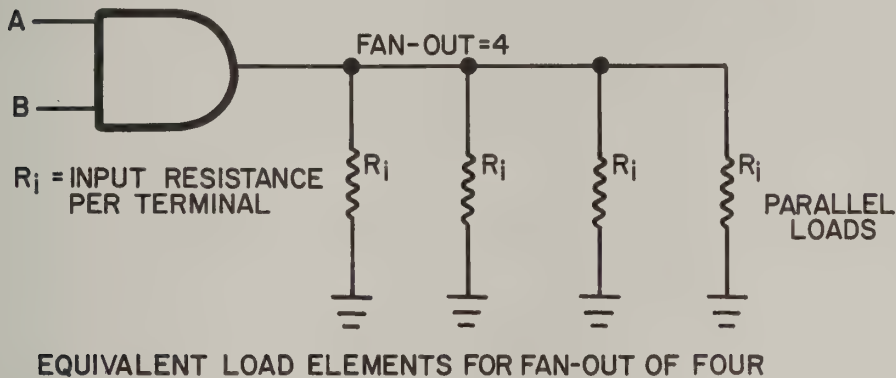


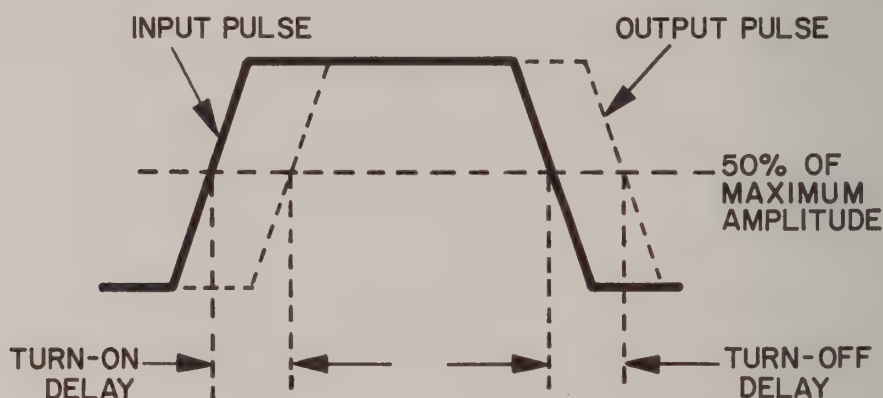
Figure 42

The FAN-OUT capability of a logical unit defines the number of **EQUIVALENT LOAD ELEMENTS** which can be driven (in parallel) by one logical output terminal, as shown in Figure 42. This approach is taken because the output of one digital IC is generally used to drive one or more inputs of several other digital IC's. An equivalent load element is defined as a resistance equal to the input resistance of one terminal of the digital circuit being rated. An IC with input terminal resistances of 480 ohms and a fan-out of 4 will drive four input terminals with a resistance of 480 ohms each or a parallel resistance of 120 ohms minimum. If the input devices being driven by the IC do not have terminal resistances of 480 ohms, the fan-out rating can still be used to determine the minimum load which can be fed by the IC.

Assume that a manufacturer makes an RTL logic circuit with all gates having an input resistance of 560 ohms with a fan-out capability of 5. Each gate is capable of driving one input terminal of five other gates, each having an input resistance of 560 ohms. The minimum load which the gate can drive is $560/5 = 112$ ohms. A problem can develop if another manufacturer's gates are substituted for the five gates driven by one of the original manufacturer's gates. If the substituted gates each have an input terminal resistance of 800 ohms, the load is $800/5 = 160$ ohms. This substitution can be easily made without overloading the original driving gate since 160 ohms is greater than the allowable minimum load (112 ohms). If, however, the substituted gates each have an input terminal resistance of 480 ohms, the load is $480/5 = 96$ ohms. This load is less than the minimum allowable; therefore, such a substitution will degrade the operation of the original driving gate. Only 4 such gates could be driven, because $480/4 = 120$ ohms; and 120 ohms exceeds the minimum allowable load.

Speed and Delay

Since almost any IC may be used in a synchronous system, a maximum clock frequency will be specified. This is the maximum gating or switching rate which the device will tolerate. In terms of bit time, the device will operate at a minimum value given by the reciprocal of the maximum clock frequency. For example, a device with a 30 MHz maximum clock frequency cannot be used in a system with a bit time less than $1/(30 \times 10^6) = 33 \times 10^{-9} = 33$ nanoseconds.



INPUT-OUTPUT WAVEFORM TIME DIFFERENCES
TWO FORMS OF PROPAGATION DELAY

Figure 43

Figure 43 illustrates the two primary **PROPAGATION DELAY TIMES** associated with a non-inverting IC gate. The **TURN-ON DELAY TIME** is measured from the instant the input signal reaches 50 percent of its maximum amplitude to the instant the output signal reaches 50 percent of its maximum amplitude, when the circuit is switching from "off" to "on." The waveform polarity is not important, although positive pulses are shown in the figure. The **TURN-OFF DELAY TIME** is also measured between the instants when the two waveforms reach 50 percent of their maximum amplitudes, when the circuit is switching from "on" to "off."

Comparisons of Logic Circuits

Table 3 provides a general comparison of most of the IC logic families described in this lesson. The table shows the reasons why DTL originally surpassed all other families and why it, in turn, is being surpassed by TTL. The DTL device offers low power consumption, good noise characteristics, and

a moderate fan-out capability along with a reasonable switching speed. The TTL device offers a better switching speed and fan-out capability with a moderate power dissipation (considerably better than CML).

Performance Characteristic	Current-Mode (ECL)	High Threshold (HTL)	Transistor-Transistor (TTL)	Diode-Transistor (DTL)	Resistor-Transistor (RTL)
Average propagation delay time (ns)	4	85	10	30	24
Binary element clock frequency, typical (MHz)	120	4	20	40	8 (min)
Power dissipation per gate (mW)	40	35	15	8	12
External noise margin	Fair	Excellent	Good	Good	Poor
System crosstalk (generated noise)	Excellent	Excellent	Fair	Good	Good
Gate fan-out	25	10	10	8	5
Available temperature range (C°) (the narrower the range, the lower the cost)	0 to +75 -55 to +125	-30 to +75	0 to +75 -55 to +125	0 to +75 -55 to +125	+15 to +55 0 to +75 0 to +100 -55 to +125
Power supply or signal voltage	-5.2 $\pm$ 20%	15 $\pm$ 1.0V	-10% 5.0 +20%	5.0 $\pm$ 10%	3.0 $\pm$ 10% 3.6 $\pm$ 10%

Comparison of Characteristics for Various Digital Logic Families

Table 3

INTEGRATED CIRCUIT FLIP-FLOPS AND LATCHES

The flip-flop or bistable multivibrator is a type of temporary memory element used in digital systems. Every bistable has two output terminals, y and $\bar{y}$. The two output terminals are always (except during transition between states) at opposite states or logic levels: 0 and 1. When the flip-flop terminal y is at a logical 0, the other terminal, $\bar{y}$, is at a logical 1; and we say that the bistable is RESET, CLEARED or in the 0 state. When y is at a logical 1

and $\bar{y}$ is at 0, the flip-flop is said to be SET or in the 1 state. This is called a temporary storage device, since a condition or state can be retained only as long as power is applied to the circuit. If the power is removed from a bistable, even for a very short interval, all of the original memory states are "lost."

There are several types of bistables. The R-S, type D, type T, and J-K are discussed in this lesson. All four of these bistables can be found in RTL, DTL, TTL, ECL, and MOS forms.

Set-Reset Flip-Flop



**THE SET-RESET (R-S)
FLIP-FLOP SYMBOL**

Figure
44

Inputs		Outputs	
R	S	y	$\bar{y}$
0	1	1	0
1	0	0	1

**Partial Truth Table
for R-S Flip-Flop**

Table
4

One basic form of bistable is the **SET-RESET (R-S) FLIP-FLOP**, shown symbolically in Figure 44. Besides the two output terminals (y and $\bar{y}$), there are also two input terminals (S and R). There are many configurations of the R-S flip-flop. The RESET (R) input is sometimes called the CLEAR input.

When a logical 1 is applied to the S input of a bistable that has been cleared, the flip-flop output goes to the logical 1 state at y and to the logical 0 state at $\bar{y}$. When a logical 1 is applied to the R input of a bistable that was set, the flip-flop output inverts, and a logical 0 appears at y with a logical 1 at $\bar{y}$. A flip-flop that is set is not affected by a set signal. When the circuit is reset, it is not affected by a clear signal. The logic assignments may be positive or negative as desired, with basic circuits composed of either NPN or PNP transistors. Based on the logical operation which has been described, a partial truth table construction for the flip-flop appears in Table 4.

~~The transistors in a flip-flop circuit may be operated in either the saturated or nonsaturated mode. For saturated operation, the transistors switch between the cutoff and saturation regions. For nonsaturated operation, the transistors switch between cutoff and a point within the active characteristic region. Saturated operation has the advantages of low power dissipation, circuit simplicity and an insensitivity to most changes in transistor operating characteristics. However, saturated switches always have a longer switching time than the unsaturated devices. The increased number of components in a nonsaturated flip-flop generally tends to raise system costs and power consumption. This is especially true in discrete circuitry, with a lesser impact on integrated circuit fabrication.~~

Since the flip-flop retains a given condition even though the initiating input pulse was removed, it constitutes a memory device. The condition of the memory can be changed by applying an appropriate pulse to the proper input at a later time.

The following Practice Exercise questions cover the subjects which you have just studied. They are:

DIGITAL IC TYPES

TRANSISTOR-TRANSISTOR LOGIC

NONSATURATED LOGIC

METAL OXIDE SEMICONDUCTOR (MOS) AND COMPLEMENTARY METAL OXIDE SEMICONDUCTOR (CMOS) LOGIC

OTHER IC LOGIC TYPES

OPERATING CHARACTERISTICS

TEMPERATURE AND DISSIPATION

VOLTAGE SWING AND NOISE

FAN-IN AND FAN-OUT

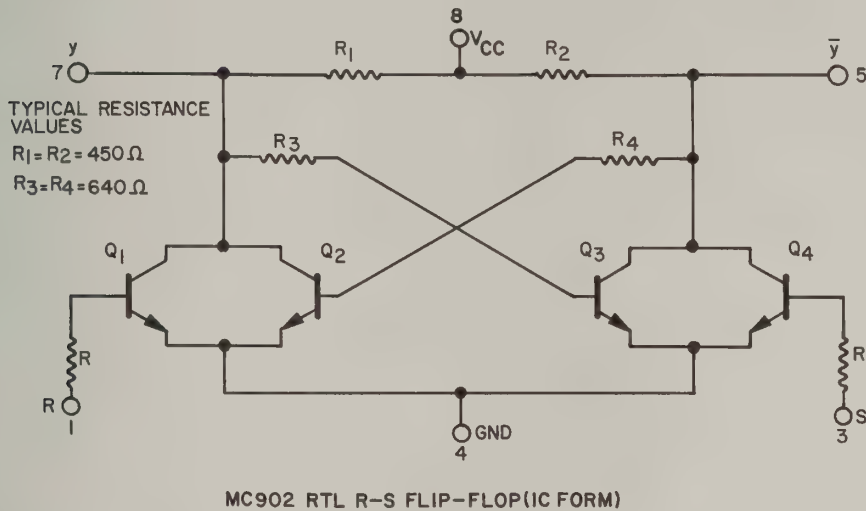
SPEED AND DELAY

COMPARISONS OF LOGIC CIRCUITS

16. Why is TTL easy to produce in IC form, but not in discrete circuit form?
17. What is the purpose of the additional input diodes in TTL using the basic circuit of Figure 26?
18. What primary advantage is offered by TTL using the basic circuit of Figure 27?
19. What precaution must be observed with unused input terminals of TTL?
20. What approach must be taken in digital IC design in order to eliminate the storage time effect?
21. What logic assignments are used with ECL circuits to obtain the OR and NOR functions?
22. What are the disadvantages of ECL?
23. What are some of the advantages of CMOS?
24. What component is used in HTL to obtain extremely high noise immunity?

25. What is the "penalty" associated with HTL?
26. What is the maximum power allowed for an IC operating at 80°C when its room temperature maximum is $350\ \mu\text{watts}$ and its derating factor is $3\ \mu\text{watts}/^{\circ}\text{C}$?
27. A particular TTL device has a dc power dissipation of $25\ \text{mW}$ and an ac power dissipation of $8\ \text{mW}$. At the operating temperature, the maximum allowable dissipation power is $30\ \text{mW}$. Can the device be used in this application?
28. A particular NAND gate has a maximum clock frequency of $25\ \text{MHz}$. What is the minimum bit time for this device?
29. Which logic types have the least critical requirements with regard to variations in supply voltage, input voltage and output voltage?
30. What is meant by the high voltage state and the low voltage state?
31. What is meant by a noise immunity of $0.8\ \text{volt}$?
32. Name two disadvantages associated with using gate expanders to increase the fan-in capability of a circuit.
33. How is an equivalent load element defined in establishing the fan-out rating of a digital circuit?
34. The output loading factor of a particular IC is 4. This output signal must be fed to three different IC's, each with an input loading factor of 1.2. Is this within the capability of the driving circuit?
35. The total propagation delay for an IC is the sum of the various delay components. True or False?

Since the initial states of a flip-flop memory are indeterminate, a pulse is applied to the RESET terminals of all elements immediately after energizing such a memory. This “clears” the memory, thus placing all flip-flops into the RESET condition.

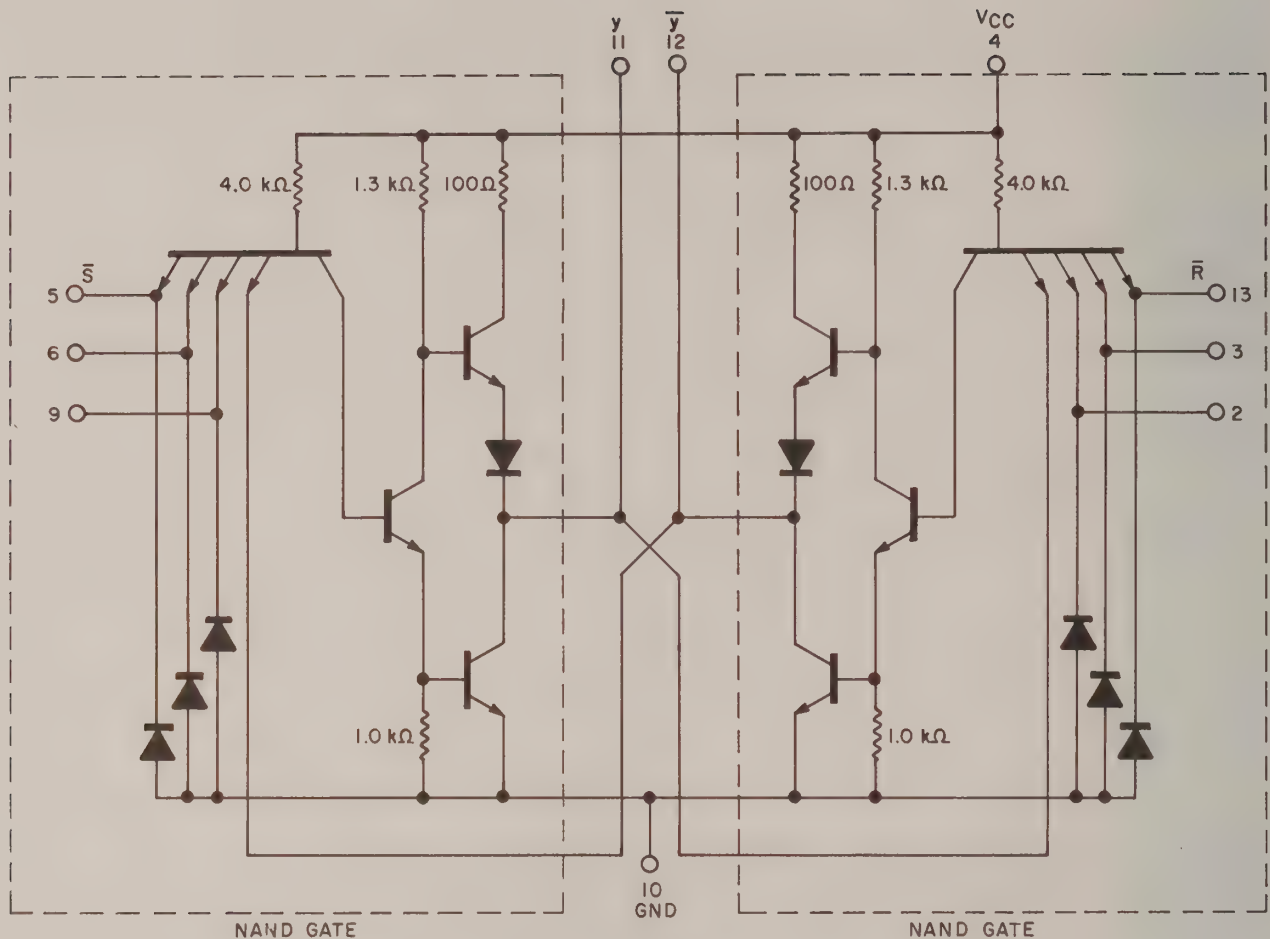


Courtesy Motorola Inc., Semiconductor Products Div.

Figure 45

A typical RTL R-S flip-flop is shown in Figure 45. This IC utilizes two transistors (operating in parallel) in the place of a single transistor. This configuration provides additional “driving” capability for the circuit, since one transistor handles the internal switching signal (the feedback from collector to base) and the other handles the input switching signals. It is not uncommon for IC’s to use two parallel transistors in the place of a single discrete-circuit transistor. The additional expense in fabrication is negligible and the matching of such transistors is excellent.

Figure 46 shows a TTL R-S flip-flop which is basically a cross coupling of two NAND gates (as indicated by the dashed line boxes). The logic diagram of the R-S flip-flop of Figure 46 is shown in Figure 47. This particular circuit is designed for operation with inverted Set ($\bar{S}$) and Reset ($\bar{R}$) input pulses (an inverted set or reset pulse is a logical 0). This means that, to clear (reset) the flip-flop, a logical 0 is applied to any or all of the inverted reset terminals (pins 2, 3 and 13), while the inverted set terminals (pins 5, 6 and 9) are all held at a logical 1. The output at terminal 11 (y) will then be a logical 0 and the output at terminal 12 ($\bar{y}$) will be a logical 1. To set the flip-flop, an inverted set pulse (logical 0) is applied to one, two, or all of the inverted set terminals, while the inverted reset terminals are all held at logical 1. The application of a logical 1 to all of the inputs (both set and reset) initiates the stable state of the flip-flop; its outputs under this condition remain in the state that they were prior to the application of the logical

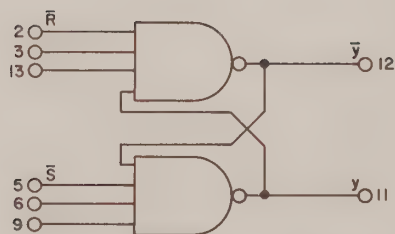


TTL R-S FLIP-FLOP CIRCUIT (MC513)
COURTESY MOTOROLA SEMICONDUCTOR PRODUCTS DIVISION

Courtesy Motorola Inc., Semiconductor Products Div.

Figure 46

1 input. Application of logical 0 to all of the inputs is not permitted, and is prevented by proper design of associated circuitry. Any unused input terminal should be tied to the supply voltage or to another input terminal of the same gate.

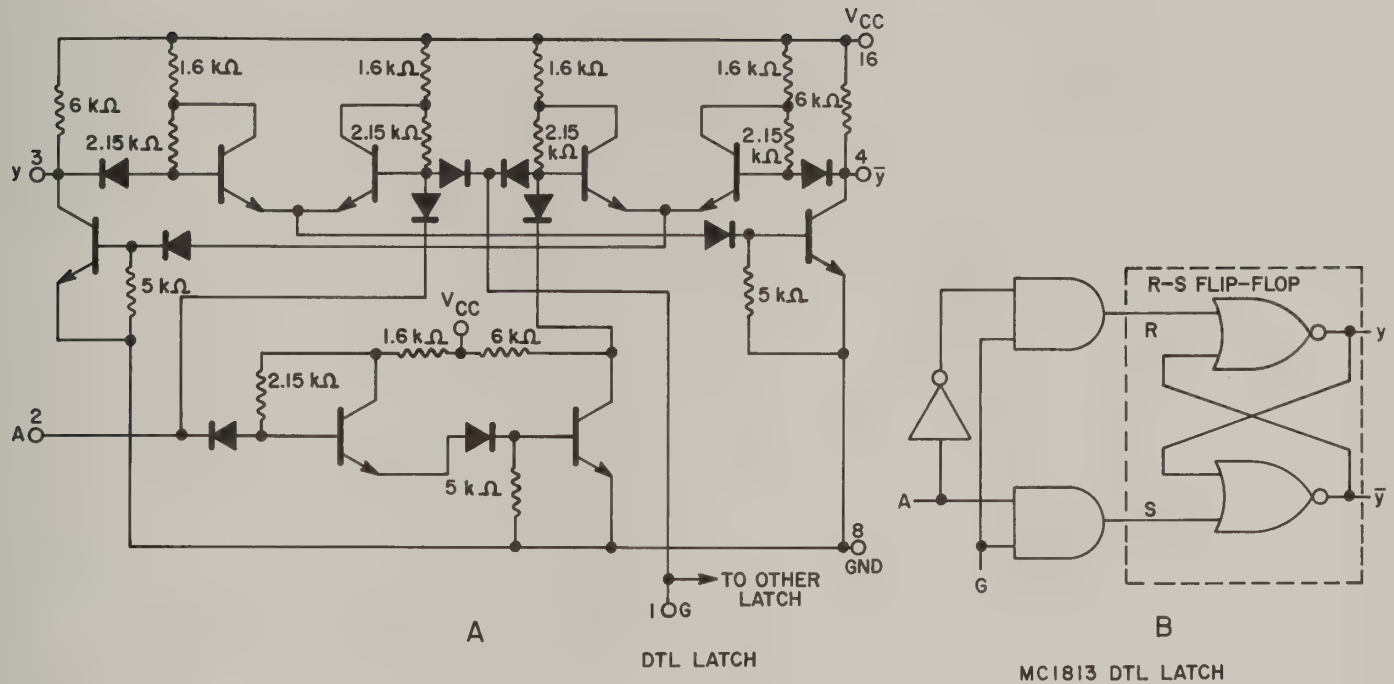


R-S FLIP-FLOP OF FIGURE 46 REPRESENTED BY NAND GATES

Figure 47

Bistable Latches

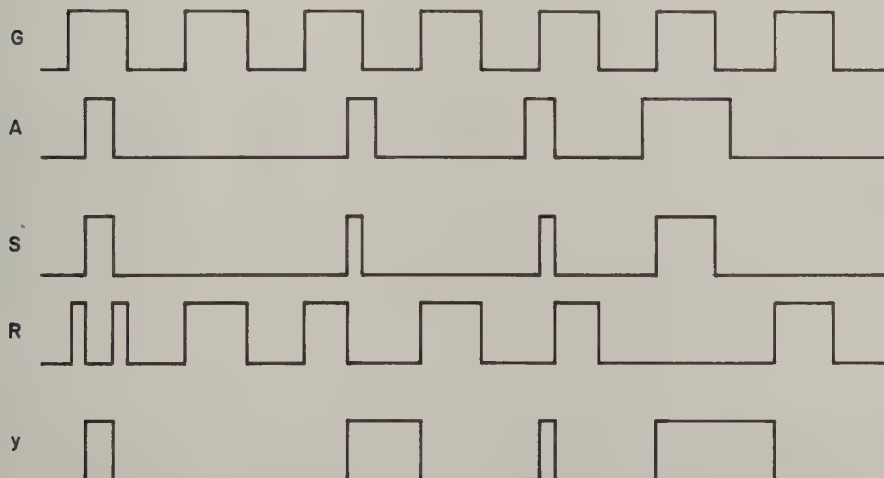
A DTL latch circuit is shown in Figure 48A, and the logic diagram of the latch circuit is shown in Figure 48B. A latch circuit is used for the temporary storage of information, with storage time dependant on certain conditions. The dashed rectangle encompasses NOR gates which form an R-S flip-flop. When the input at G (called the clock or enable input) is low (a logical 0), both of the AND gates are disabled, preventing the flip-flop from changing states. When the input at G is high (a logical 1), the two AND gates are



Courtesy Motorola Inc., Semiconductor Products Div.

Figure 48

enabled, and the y output of this flip-flop will then be at the same logic level as the signal input at terminal A, as shown by the waveforms of Figure 49. When terminal A (the signal input) is at logical 1, the output of the bottom AND gate will be at a logical 1, causing the flip-flop to set. When terminal A is at logical 0, the output of the top AND gate will be at a logical 1, resetting the flip-flop.



TIMING DIAGRAM FOR THE CIRCUIT OF FIGURE 48

Figure 49

A bistable latch employing MOSFET's is shown in Figure 50. This circuit requires both logical 1 and logical 0 clock inputs (G and $\bar{G}$) for proper operation. When G is at a logical 1, Q_1 transfers input information from terminal A to Q_2 . Q_3 is cut off by a logical 0 at terminal $\bar{G}$. Q_2 and its pull-up resistor and Q_4 and its pull-up resistor form inverter circuits. When terminal G is at logical 0, Q_1 becomes cut off and Q_3 conducts, clamping the gate of Q_2 to the output. This action now prevents the output, y , from following the input at terminal A . As in the other MOS circuits, the pull-up resistors can be replaced by special MOSFET's.

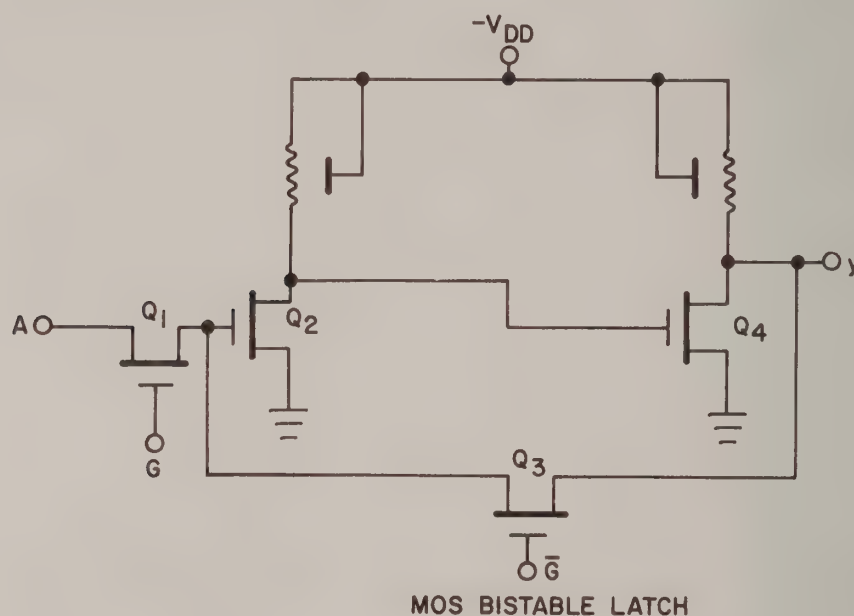


Figure 50

Type-D Flip-Flops

In (t)	Out (t+)	
A	y	$\bar{y}$
1	1	0
0	0	1

Truth Table for
Type D Flip-Flop

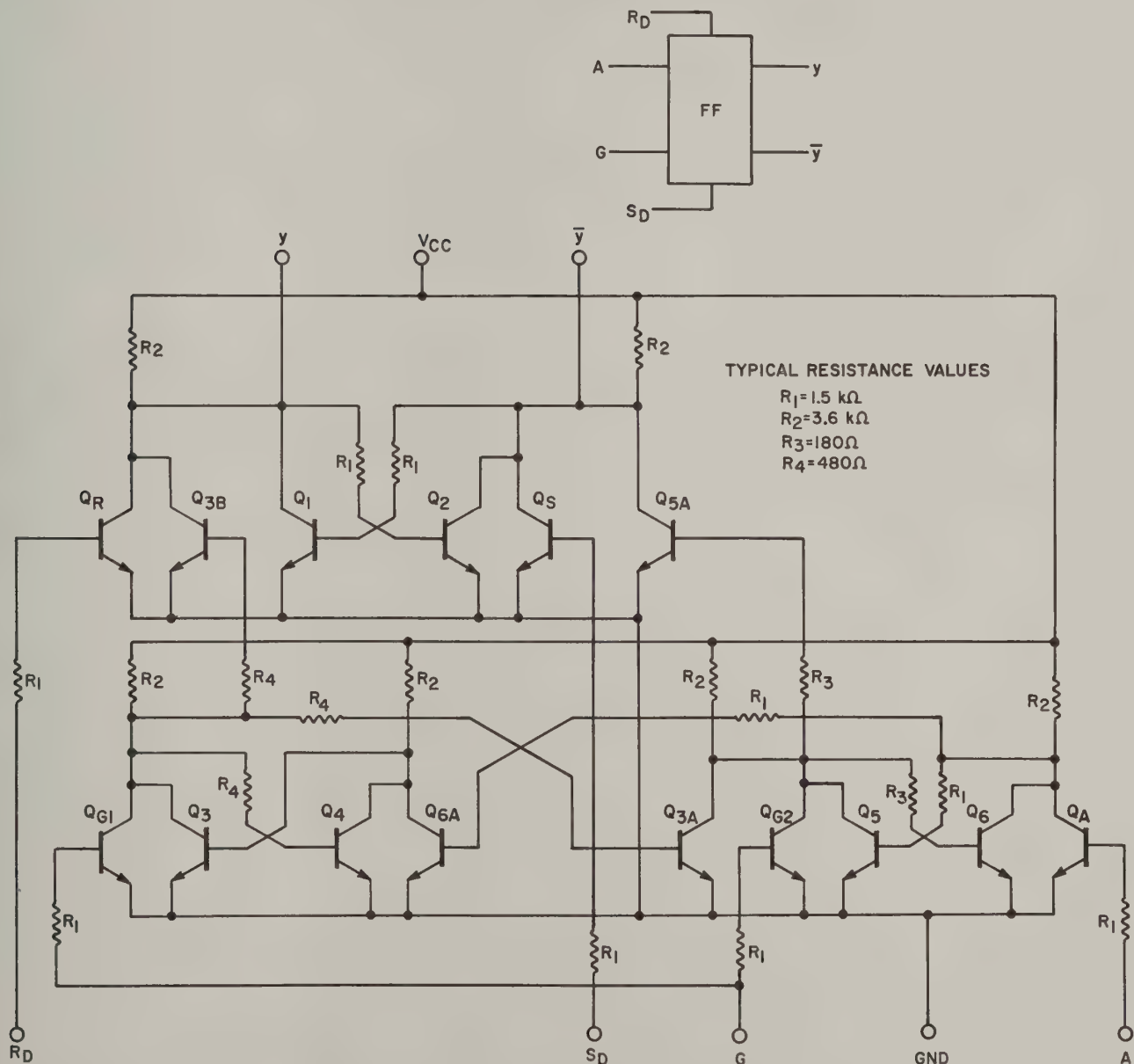
Table
5

The basic R-S flip-flop responds to two logic signals: the SET and the RESET inputs. When used as a temporary storage element, the logic level at one input is the "information" to be stored. When another input is required to synchronize the flip-flop with other units in a sequential system to assure that the information enters the flip-flop at the right time, the device becomes a **GATED FLIP-FLOP**. This synchronization might be accomplished through the use of timing signals from a master clock or other signals especially generated for this purpose. Gated flip-flops are available in RTL, DTL, or TTL integrated circuits.

The **TYPE-D FLIP-FLOP** is a common type of gated arrangement. In this form, the application of the gate voltage delays the logic value at terminal A , holding it for the duration of the existing gate pulse. The truth table for this operation appears as Table 5. As shown, the value at terminal A at a particular time interval t does not act on the output until the next gating state ($t+$). After the switching occurs, a change in signal level at terminal A will not be

reflected in the output until the next gating state. The signal at terminal A may be of very short duration; it may even be removed before the gating state changes. The effect will be the same since the circuit "holds" the value until the proper time. The gating signal is not shown in the truth table since it only determines when the signal levels enter the flip-flop and therefore it has no logical effect on the output.

DTL CIRCUIT USED FOR MC1906 TWO-INPUT QUAD AND GATE
COURTESY MOTOROLA SEMICONDUCTOR PRODUCTS DIVISION



MC913 RTL TYPE-D FLIP-FLOP (IC FORM)
COURTESY MOTOROLA SEMICONDUCTOR PRODUCTS DIVISION

Courtesy Motorola Inc., Semiconductor Products Div.

Figure 51

Figure 51 shows the schematic of a type-D flip-flop integrated circuit. This is an RTL device utilizing three flip-flop stages to accomplish the input signal holding action. Q_1 and Q_2 form the output stage; Q_3 and Q_4 form the intermediate flip-flop; Q_5 and Q_6 form the input stage. The arrangement is such that the signal level at A changes the output flip-flop state only during NEGATIVE TRANSITIONS of the gate voltage. That is, if the y output is not the same as the signal at terminal A, the value of the y output will change the next time the gate voltage goes from a high positive value (logical 1) to a low positive value (logical 0). No other condition, including the POSITIVE TRANSITION of the gate voltage, has any effect on the output of the type-D flip-flop.

This is further illustrated in Figure 52. The clock signal is G, and the data input signal is A. Figure 52 illustrates three data input conditions: data pulses 1, 2 and 3. Data pulse 1 goes to a logical 1 and back to a logical 0

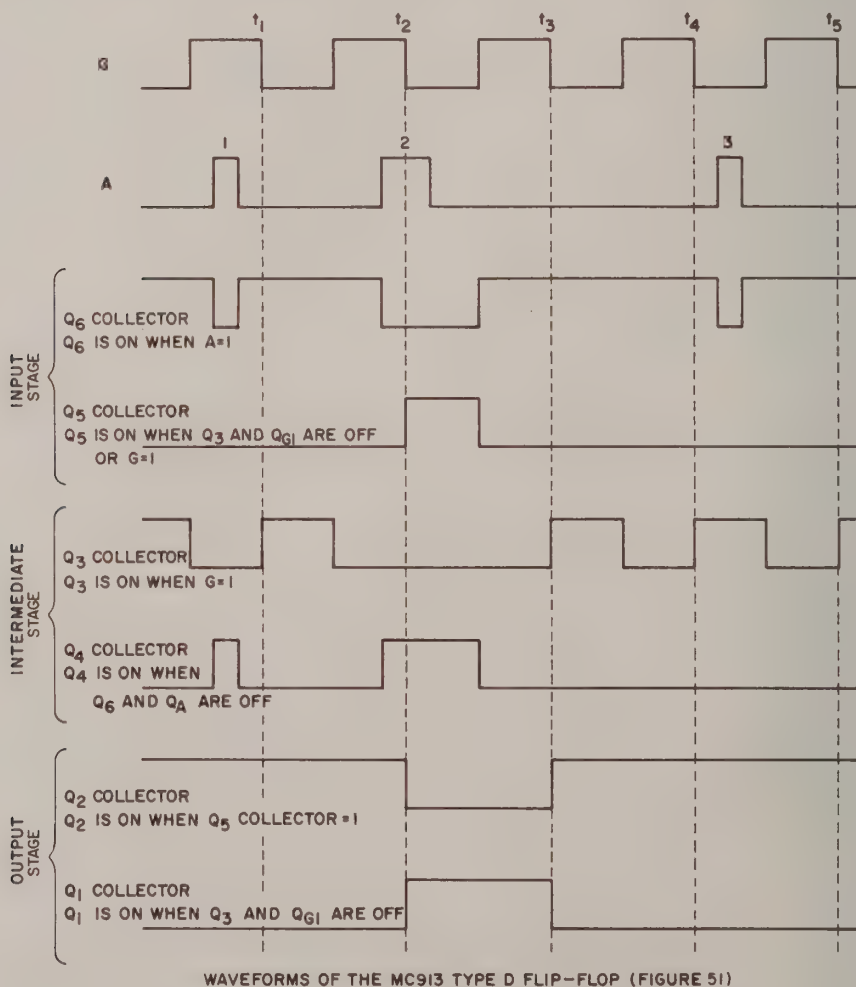


Figure 52

while G is at logical 1. Data pulse 2 goes to a logical 1 while G is at a logical 1, returning to a logical 0 after G returns to logical 0. Data pulse 3 goes to a logical 1 and returns to a logical 0 while G remains at a logical 0. Q_{G1} and Q_{G2} saturate when G is at logical 1. Q_{6A} saturates when the collector of Q_6 is at logical 1.

Initially, the type-D flip-flop shown in Figure 51 is cleared; Q_1 is conducting and Q_2 is cut off. Q_A saturates whenever terminal A is at 1. Q_{G1} and Q_{G2} saturate when the clock is at logical 1. When the collector of Q_5 is at logical 0, caused by the saturation of Q_5 , Q_{G2} , or Q_{3A} , the Q_6 collector will be at logical 1 (unless Q_A is saturated). Therefore, the collector of Q_5 goes to logical 1 only when terminal A is logical 1 as G goes to logical 0 (such as at t_2). When the collector of Q_5 is a logical 1, Q_{5A} saturates, making $\bar{y}$ a logical 0. This cuts off Q_1 , making y a logical 1. The flip-flop is now set and it remains set until the clock goes from logical 1 to logical 0 again when A is logical 0 (at t_3). Figure 52 shows that the flip-flop is cleared at t_3 because Q_3 and Q_{G1} both become cut off. The collector of Q_3 becomes a logical 1 because the collector of Q_4 is at a logical 0.

The circuit shown in Figure 51 has one additional feature: **DIRECT SET** and **DIRECT RESET** terminals, denoted by S_D and R_D , respectively. These inputs allow the circuit to be operated as a conventional R-S flip-flop when the gate input is at logical 1. Such a combination of flip-flop options is frequently included in IC's, since the additional fabrication costs are negligible.

Type-T Flip-Flops

The **TYPE-T FLIP-FLOP** operates as shown in the logic diagrams of Figure 53. The inverse of the output is gated into the input; however, it is delayed by one gating interval. With this arrangement, the output state of the flip-flop changes each time a particular gating signal condition occurs. A typical input-output timing diagram is shown in Figure 53B. This diagram is associated with a circuit which switches on the positive gate transition. Each time the gate level changes from a low value to a high value (from logical 0 to logical 1), the output state changes. The effect is to produce an output whose frequency is half of the gating frequency.

The operation of the type-T flip-flop can be explained by examining the logic of Figure 53. With the output level at logical 0 ($y = 0$ as shown in Figure 53A), the inverse (logical 1) is fed back from the $\bar{y}$ output to the flip-flop input and held for one gating interval. This delayed signal is applied to the top AND gate at A' , and is inverted, enabling a logical 0 to be applied to the bottom AND gate at A'' . As long as the gate signal is a logical 0, neither AND gate will switch the flip-flop. However, as soon as the next logical 1 occurs on the gate, the top AND gate applies a logical 1 to the SET terminal of the flip-flop, causing it to switch (Figure 53C).

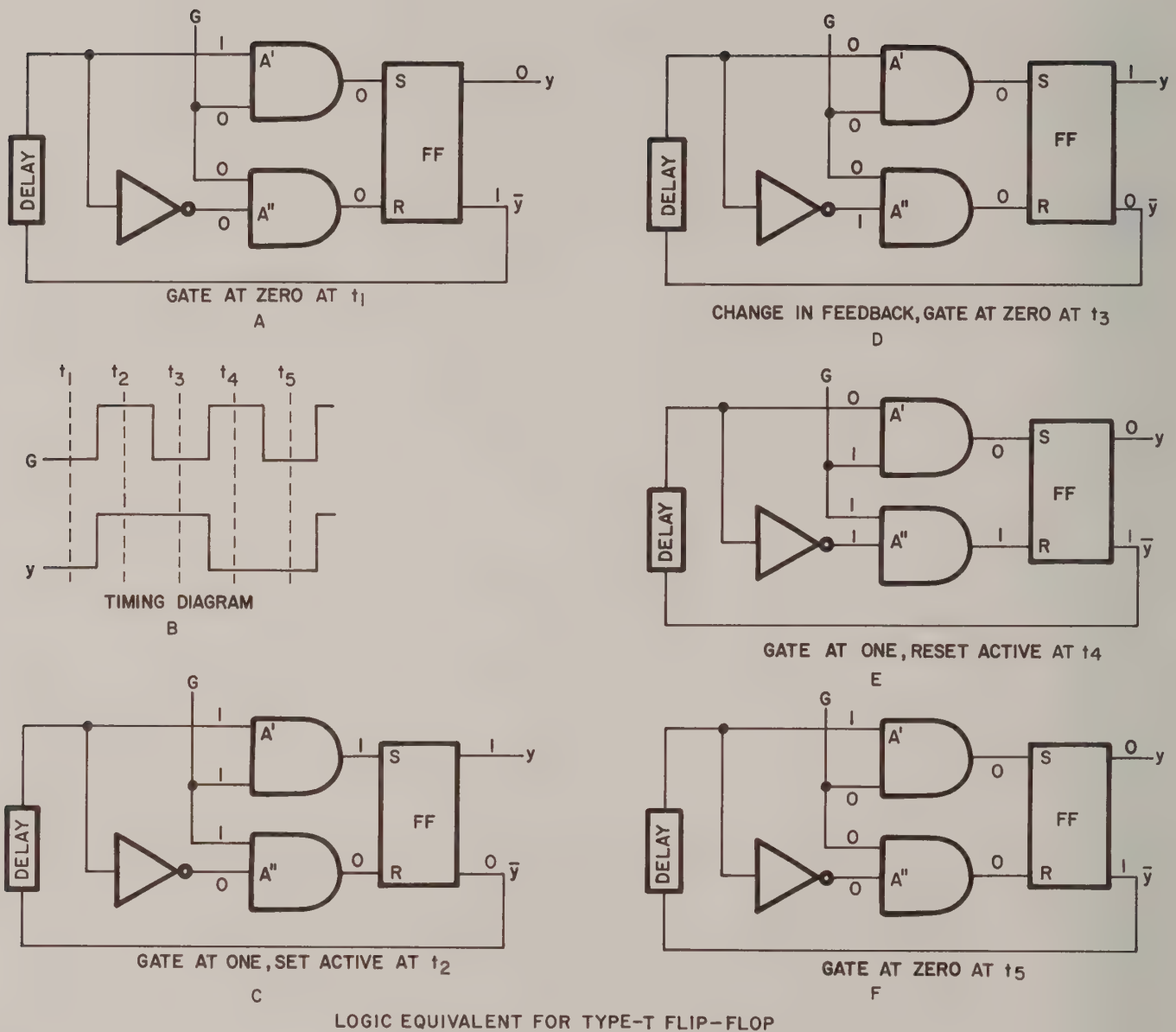


Figure 53

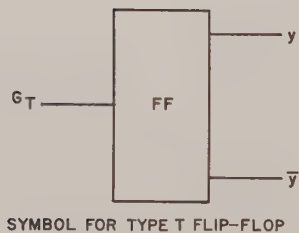


Figure 54

This causes the feedback signal to change, as shown in Figure 53D. A logical 0 now appears at A' , and a logical 1 at A'' . With the gate signal back at logical 0, no change can occur in the flip-flop (neither AND gate affects the flip-flop). When the next logical 1 occurs on the gate input, the lower AND gate applies a logical 1 to the RESET terminal of the flip-flop, causing it to switch (Figure 53E). A logical 0 is fed back to input A'' and the circuit conditions return to those shown in Figure 53A. This cyclic operation continues as long as the gating signal is applied. The symbol for the type-T flip-flop is shown in Figure 54.

The J-K Flip-Flop

The **J-K FLIP-FLOP** is a circuit which is a refinement of the R-S flip-flop. The J-K flip-flop also incorporates both the type-D and type-T actions. This flip-flop has become a predominant unit in IC form, especially when the J-K arrangement includes stages with **DIRECT SET** and **DIRECT RESET** capabilities. Such an arrangement can accomplish every type of flip-flop action that has been described in this lesson. The J-K flip-flop can also be operated asynchronously through an appropriate selection of the input terminals. The symbol for a J-K flip-flop is shown in Figure 55. The J-K flip-flop truth table is essentially the same as the R-S flip-flop truth table except for the condition when both inputs are logical 1 simultaneously. This condition always causes the J-K outputs to change states.

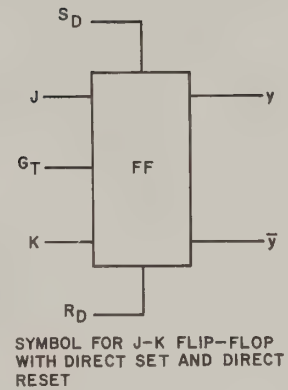
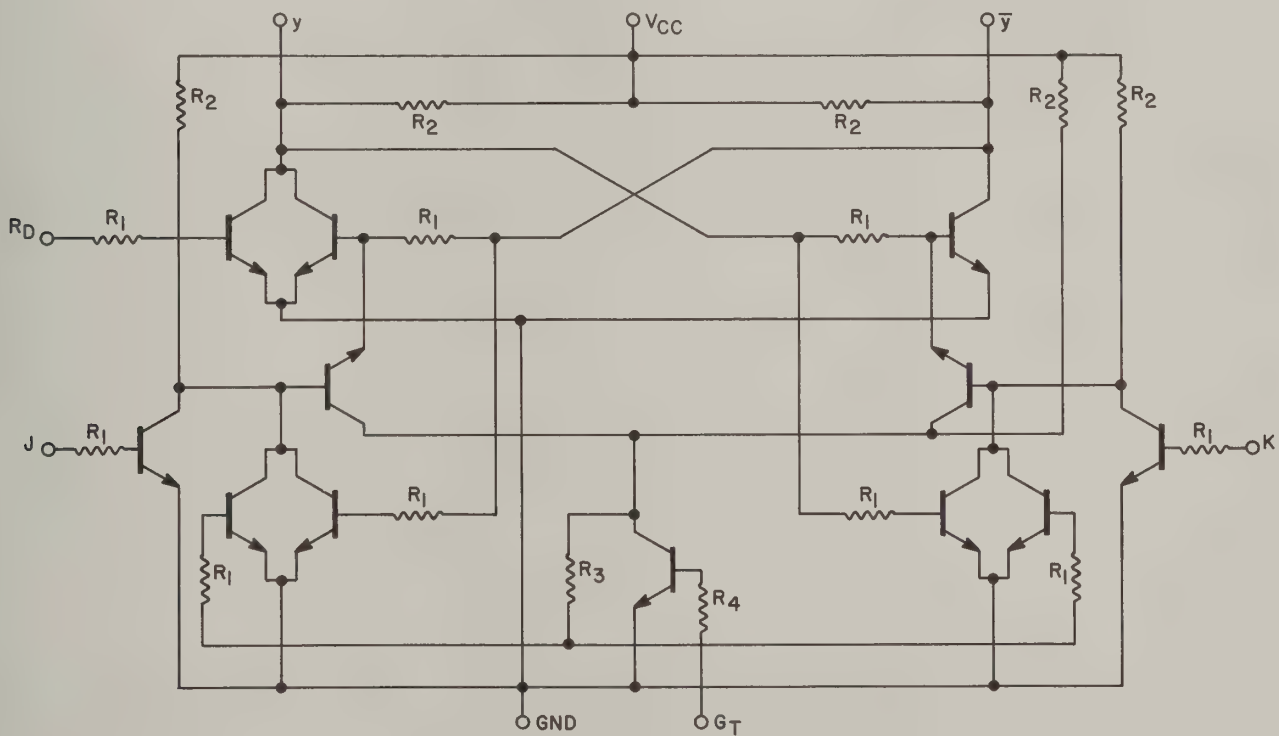


Figure 55



TYPICAL RESISTANCE VALUES

$$R_1 = 450 \Omega$$

$$R_2 = 640 \Omega$$

$$R_3 = 510 \Omega$$

$$R_4 = 225 \Omega$$

MC916 RTL J-K FLIP-FLOP CIRCUIT (IC FORM)

Courtesy Motorola Inc., Semiconductor Products Div.

Figure 56

One of the simpler J-K flip-flops in IC form is the RTL unit shown in Figure 56. This unit incorporates the DIRECT RESET capability, but does not include the DIRECT SET capability. The R_D terminal is used to clear all elements in a temporary storage system. This also resets all logic levels in the system to the desired initial condition levels. It can also be used for this purpose any other time the memory is to “forget” all previous conditions and start “fresh.” Operated independently, a logical 1 at either the J or K inputs causes the flip-flop to switch states. With a logical 1 at both J and K simultaneously (usually achieved by tying J and K together) the circuit operates as a type-T flip-flop. With the J and K terminals always at opposite states (achieved by placing an inverter between J and K) and a control signal on the G input, the circuit produces type-D operation.

SUMMARY

The NOR, NAND, and NOT functions are all based on the use of the CE amplifier configuration. By using the CE stage as a common design, IC's can be developed and manufactured to satisfy almost any logic function requirement without designing new circuitry for each application. When the OR function is desired, an existing NOR design can be followed by the basic CE stage (NOT function). When the AND function is desired, the basic NAND design is followed by a NOT stage. This approach has helped to minimize IC costs.

IC chips can be packaged in TO cans similar to those used for transistors, in flat metal or ceramic packages, or in a plastic encapsulation. These packages provide physical and mechanical protection for the chip, and support the pins and leads used for electrical interconnection. Plastic packages are the least protective. However, they are adequate for most industrial applications.

Thin-film IC's use discrete transistors and diodes; however, all resistors, capacitors and interconnecting leads are evaporated onto a glass or ceramic substrate. These IC's provide the greatest precision in component values. Monolithic IC's are “truly” integrated circuits. All elements (transistors, diodes, resistors, etc.) are formed within the semiconductor substrate material by appropriately forming P-type and N-type regions and PN junctions. Interconnections are formed with thin metallic layers similar to those on the thin-film IC's.

Modern technology allows many thousands of circuit elements to be formed within a single IC chip. The development of MOSFET IC's has increased this component density significantly. Newer techniques are constantly being developed.

The following Practice Exercise questions cover the subjects which you have just studied. They are:

**INTEGRATED CIRCUIT FLIP-FLOPS AND
LATCHES**

SET-RESET FLIP-FLOP

BISTABLE LATCHES

TYPE-D FLIP-FLOPS

TYPE-T FLIP-FLOPS

THE J-K FLIP-FLOP

36. The basic R-S flip-flop has two output terminals and two input terminals. True or False?
37. What are the respective states of the two output terminals of a basic flip-flop, regardless of the input conditions?
38. The outputs of a particular flip-flop are $y = 0$ and $\bar{y} = 1$. Which input terminal must have a logical 1 applied in order to change this condition?
39. Why is the flip-flop a memory device?
40. Why is a pulse applied to the RESET terminals of all elements of a flip-flop memory immediately after the memory is energized?
41. When multiple inputs are used to SET or RESET the flip-flop of Figure 46, the switching action will occur only if all of the appropriate inputs occur simultaneously. True or False?
42. Justify the switching action of cross-coupled NAND gates of Figure 47 when the initial outputs are $\bar{y} = 1$ and $y = 0$.
43. A bistable latch can be composed of an inverter circuit, two AND gates and an _____.
44. What is a gated flip-flop?
45. What is unique about the operation of the type-D flip-flop?
46. Why is it unnecessary to show the gate condition in the truth table for a gated flip-flop?
47. A type-D flip-flop which switches on negative transitions of the gate signal

will change states only when the gate voltage drops to a logical 0 if y and A are different than at the last negative transition of the gate. True or False?

48. What is the numerical relationship between the output and gating frequencies of the type-T flip-flop?
49. Explain how the logic network in Figure 53 switches when y is initially 1 and the gate signal goes from 0 to 1.
50. What types of operation can be accomplished by a J-K flip-flop with separate provisions for a DIRECT SET and DIRECT RESET input?
51. What is the DIRECT RESET input of the unit in Figure 56 (or any other similar unit) used for?
52. How is type-D operation accomplished with the J-K flip-flop?

Monolithic digital IC's utilize many of the basic logic arrangements used in discrete circuitry. Examples include resistor-transistor logic (RTL) and diode-transistor logic (DTL). The basic RTL IC function is NOR and OR (which is the NOR-NOT combination). NAND and AND are available in only limited numbers, since they require complex RTL arrangements. An inversion of the logic level assignments for NOR and OR IC's produces the dual functions, NAND and AND. The basic DTL IC functions are NAND and AND. However, a good variety of NOR and OR IC's is also available in this logic family.

Transistor-transistor logic (TTL) circuits are unique to IC's. Special transistors are formed with multiple-input emitter terminals. Such an approach would require separate transistors in a discrete circuit.

RTL, DTL, TTL, and MOSFET logic IC's operate in the saturated mode. IC's that operate in the unsaturated mode are also available. These utilize current mode logic (CML) arrangements, with the predominant form known as emitter coupled logic (ECL).

Other specialized IC families include high threshold logic (HTL) in either the RTL, DTL, or TTL forms (HRTL, HDTL or HTTL). These forms utilize zener diodes to raise the switching thresholds. Complementary transistor logic (CTL) utilizes both NPN and PNP transistors on the same chip; however, it is not in common usage.

Generally, IC's are low-power devices, operating in the milliwatt range or below. Important operating characteristics include the maximum rated power, maximum clock frequency, switching thresholds, minimum logical 1 and maximum logical 0 input voltages, noise margins, fan-in and fan-out (loading capability), and propagation delay times.

HTL circuits have long propagation delays; however, they also have the greatest noise immunity. ECL circuits provide the greatest switching speed and the smallest propagation delay. TTL and DTL are the most popular "compromise" families. All are subject to temperature variations and require fairly "tight" control of their power supply voltage.

IC flip-flops and latches are important forms of temporary memory in digital systems. IC flip-flops are available in RTL, DTL, TTL, and ECL forms. They are frequently manufactured with integral multiple-input AND gates, enabling several simultaneous conditions to be required to SET or RESET the device.

Gated flip-flops, which change the logic levels of their outputs when certain gating conditions occur, are available in latches, type-D, and type-T forms. Bistable latches allow the output to follow the input under one clock or gate condition and hold the output under the other gate condition. Type-D devices hold an input condition which exists at a particular gating condition; and then pass the inputs to the flip-flop when the next gate occurs. Type-T devices change their output states each time a particular gate condition occurs.

A unique IC flip-flop is the J-K device. It can be used as an R-S flip-flop, as a type-D flip-flop, or as a type-T flip-flop, by changing the arrangement of its input connections.

IMPORTANT DEFINITIONS

ASYNCHRONOUS—A mode of digital operation where sequential logic decisions are initiated only when the previous decision is complete. There is no “fixed” time sequencing of events. Opposite of SYNCHRONOUS.

BIT-TIME—The time allowed for each decision in a synchronous system.

BONDING PADS—Comparatively large metal film areas on an IC chip which are used to connect lead wires from the chip to the package pins.

BUFFER—A stage or gate providing high output power to drive subsequent circuitry.

CHIP—One of many identical IC's formed on a single piece of substrate material. After fabrication, the chips are separated and mounted in the desired TO, flat, or plastic packages.

COMPLEMENTARY TRANSISTOR LOGIC (CTL)—Logic circuitry utilizing both NPN and PNP transistors.

DARLINGTON CONFIGURATION—A two-transistor arrangement (of similar types) with collectors connected together and the emitter of the input transistor connected to the base of the second transistor. It is characterized by a high current gain and a high input resistance.

DIFFERENTIAL AMPLIFIER—A bridge arrangement of transistors and their respective loads. The output is normally taken from between the two collectors. Operation may completely cancel the differences between two input signals, or amplify the difference between two input signals.

DIRECT RESET—An arrangement used with TYPE-D, TYPE-T, or J-K FLIP-FLOPS to change the y output to 0 and the y output to 1 in an ASYNCHRONOUS manner.

DIRECT SET—An arrangement used with TYPE-D, TYPE-T, or J-K FLIP-FLOPS to change the y output to 1 and the y output to 0 in an ASYNCHRONOUS manner.

EQUIVALENT LOAD ELEMENTS—Equivalent resistances used to rate the output capability of logic circuitry. One element is identical to the input resistance (one terminal) of the circuit being rated.

EXPANDABLE GATE—A gate which may have its FAN-IN capability increased by external circuit connections.

FAN-IN—A specification of the number of available input conditions which may drive a digital circuit.

IMPORTANT DEFINITIONS (Continued)

FAN-OUT—A specification of the number of EQUIVALENT LOAD ELEMENTS which a digital circuit may drive.

GATED FLIP-FLOP—A flip-flop whose switching operation is controlled by signals other than the information to be stored.

HIGH THRESHOLD LOGIC (HTL, HDTL, or HTTL)—Logic circuits designed to have especially high SWITCHING THRESHOLDS and, therefore, a very high NOISE IMMUNITY.

INSULATING SUBSTRATE—A ceramic or glass material upon which the components of a THIN-FILM IC are deposited. Also, the basic N- or P-type material upon which MONOLITHIC IC's are "built."

J-K FLIP-FLOP—A flip-flop capable of both TYPE-T and TYPE-D operation.

LARGE-SCALE INTEGRATION (LSI)—The electrical interconnection of large quantities of active and passive components on the face of a single IC chip. Sometimes referred to as MEDIUM-SCALE INTEGRATION.

LOGIC FAMILY—A particular grouping of digital circuits, classified on the basis of circuit components or operating mode.

LOGIC SYSTEM—See LOGIC FAMILY.

MAXIMUM 0 INPUT VOLTAGE—The maximum logical 0 voltage level which may be applied to a digital circuit without causing it to switch states.

MEDIUM-SCALE INTEGRATION (MSI)—The interconnection of electrical components on a single IC chip in somewhat smaller quantities than with LARGE-SCALE INTEGRATION.

MINIMUM 1 INPUT VOLTAGE—The minimum logical 1 voltage level which may be applied to a digital circuit without causing it to switch states.

MONOLITHIC IC—An IC which has all components fabricated on a single piece of semiconductor material.

NEGATIVE TRANSITIONS—Changes in an electrical waveform from a high to a low value.

NOISE MARGINS—Differences between the HIGH VOLTAGE STATE and the MINIMUM 1 INPUT VOLTAGE, or between the MAXIMUM 0 INPUT VOLTAGE and the LOW VOLTAGE STATE.

IMPORTANT DEFINITIONS (Continued)

The two values are not necessarily the same. The smaller of the two is used to define the noise immunity of the circuit.

PARASITIC CAPACITANCES—Stray capacitances in integrated circuits, particularly at PN junctions in the diode and transistor elements.

POSITIVE TRANSITION—A change in an electrical waveform from a low to a high value.

PROPAGATION DELAY TIMES—The time delay between the instant when the input pulse reaches 50% of its maximum value and the instant when the output pulse reaches 50% of its maximum value. See **TURN-ON** and **TURN-OFF DELAY TIMES**.

SET-RESET FLIP-FLOP—The basic flip-flop arrangement with two input trigger terminals; one to **SET** the flip-flop, the other to **RESET** the flip-flop.

SWITCHING THRESHOLDS—Minimum and maximum signal values which cause a change in the operating condition of a digital circuit.

SYNCHRONOUS—Sequences of events which are in step time-wise. Opposite of **ASYNCHRONOUS**.

THIN-FILM IC—An IC which utilizes thin-film patterns of metals and dielectrics to form passive circuit components.

TRANSISTOR-TRANSISTOR LOGIC (TTL)—Logic circuitry utilizing separate transistors (or emitter elements, in IC's) for each input condition; all the inputs then feed to another transistor switching circuit.

TURN-OFF DELAY TIME—The delay from the trailing edge 50% point of the input pulse to the 50% point of the trailing edge of the output. A form of **PROPAGATION DELAY**.

TURN-ON DELAY TIME—The delay from the 50% point of the input pulse leading edge to the 50% point of the output pulse leading edge. A form of **PROPAGATION DELAY**.

TYPE-D FLIP-FLOP—A **GATED FLIP-FLOP** arrangement which delays the logic value at the input for one gating interval, then allows the circuit to respond to this condition.

TYPE-T FLIP-FLOP—A **GATED FLIP-FLOP** which changes output conditions each time a particular gating condition occurs. This is accomplished by inverting the output and feeding this signal to the input one gating interval later. This produces an output frequency which is one-half the gating frequency.

ESSENTIAL SYMBOLS AND EQUATIONS

$$\overline{\overline{A + B}} = A + B \quad (1)$$

$$\overline{\overline{A \cdot B}} = A \cdot B \quad (2)$$

$$1 \text{ noise margin} = \text{normal } 1 \text{ voltage} - \text{minimum } 1 \text{ voltage} \quad (3)$$

$$0 \text{ noise margin} = \text{maximum } 0 \text{ voltage} - \text{normal } 0 \text{ voltage} \quad (4)$$

PRACTICE EXERCISE SOLUTIONS

1. The NAND function, $\overline{A \cdot B \cdot C}$, can be followed by a NOT function to produce the AND function, $A \cdot B \cdot C$.
2. The NOR and NAND functions are based on the CE amplifier arrangement, as in the NOT function. By combining NOR and NOT, the OR function is obtained. By combining NAND and NOT, the AND function is obtained. Therefore, the single basic CE design can be used to obtain all the basic functions, minimizing IC engineering and development costs.
3. The IC package provides adequate environmental and mechanical protection, provides support for the interconnecting leads, and serves as a heat sink.
4. Plastic does not serve as a heat sink as efficiently as metal or ceramic, and it also does not provide the same environmental and mechanical protection. For example, plastic IC's are not hermetically sealed as are metal and ceramic units.
5. The monolithic IC capacitor is a large area P-N junction.
6. Parasitic capacitances are capacitances which are present at every P-N junction of every component on a monolithic chip.
7. False
8. opposite
9. The dual function for any given circuit can be obtained by inverting the normal logic assignments for the circuit.
10. The logic $\overline{A + B + C + D} = Z$ is the dual of $\overline{A \cdot B \cdot C \cdot D} = Z$. Therefore, it can be obtained by inverting the logic assignment for the circuit.
11. RTL IC's have been in use for so long that they are the least expensive digital IC available. If performance is not of special concern, the RTL IC can still be used.
12. False
13. One input terminal is arranged to enable several separate inputs to be attached, thereby increasing the input capability.
14. When a basic NOR or NAND circuit design is followed by an additional CE stage (NOT circuit), the OR or AND functions are produced.
15. Separate CC stages are required for each input of a DTL OR/NOR circuit, while only one such stage is used for AND/NAND circuits, regardless of the number of inputs.

PRACTICE EXERCISE SOLUTIONS (Continued)

16. The large number of transistors required are easily fabricated in IC form at a very low cost; however, in discrete form, individual transistors increase circuit cost significantly.
17. Additional input diodes in TTL provide a clamping action, preventing undershoot during the turn-off operation.
18. The primary advantage offered by TTL is that it provides a low output impedance, reducing turn-off time to near the theoretical limit for saturated switching.
19. These terminals should be returned to the supply voltage or tied to the "used" terminals in order to avoid the introduction of external noise signals.
20. Unsaturated switching must be used.
21. Basically a positive logic, with a low negative level for logical 1; the high negative level for logical 0.
22. The disadvantages of ECL are low noise immunity and increased circuit complexity due to tapped-supply or multiple-supply requirement (at least one supply voltage is needed, as well as a reference voltage or a circuit for the reference voltage).
23. Very low power dissipation (current is drawn only when switching takes place) and operation is possible with supply voltages between 3 and 15 volts.
24. The zener diode is used in HTL to obtain extremely high noise immunity.
25. Very slow switching time is the "penalty" associated with HTL.
26. $350 - 3(80 - 25) = 350 - (3)(55) = 350 - 165 = 185 \mu\text{W}$.
27. The total dissipation is $25 + 8 = 33 \text{ mW}$. This exceeds the maximum allowable (30 mW) and therefore the IC cannot be used.
28. $1/(25 \times 10^6) = 0.04 \times 10^{-6} = 40 \text{ nanoseconds}$.
29. TTL and ECL.
30. The high voltage state is the voltage normally used for logical 1; the low voltage state is the voltage normally used for logical 0.
31. The peak noise amplitude in the circuit must exceed 0.8 volt in order to cause erroneous switching.

PRACTICE EXERCISE SOLUTIONS (Continued)

32. The minimum 1 input voltage increases and the maximum 0 input voltage decreases. This requires more driving units and lowers the 0 noise margin.
33. The equivalent load element is the same resistance value as the input resistance of the device being rated.
34. The total units are $3(1.2) = 3.6$. This is less than 4 and is therefore within the capability of the circuit.
35. True
36. True
37. The two output terminals are at opposite states.
38. The SET terminal must have a logical 1 applied to it in order to change the y output from logical 0 to logical 1 and the $\bar{y}$ output from logical 1 to logical 0.
39. A flip-flop is a memory device because its output conditions remain unchanged until a new input is applied, as long as the system remains energized.
40. To make all the elements in the same initial state (Clear) a pulse is applied to the RESET terminals of all elements of a flip-flop memory immediately after the memory is energized.
41. False—Only one input at the NAND gate whose output is at 0 needs to be switched to 0 (the inputs to the other NAND gate must all be at 1).
42. With $\bar{R} = 0$ and $\bar{S} = 1$, the lower NAND gate output is a logical 1, changing y from 0 to 1. This change is fed back to the input of the upper NAND gate, forcing its output to 0.
43. R-S flip-flop.
44. A gated flip-flop is operated under the controlling influence of a special signal (usually the timing signal).
45. The input condition on a type D flip-flop will not change the flip-flop condition until the NEXT change of state of the gate signal occurs.
46. The gate controls the time that the decision will be made; it does not affect the logic of the operation.
47. True

PRACTICE EXERCISE SOLUTIONS (Continued)

48. The output frequency is one-half the gating frequency.
49. With y initially equal to 1, a 0 is applied to the feedback delay unit. One time interval later, this 0 is applied to the upper AND unit at A' , and a 1 is applied to the lower AND unit at A'' . If the gate signal goes to 1 at this time, the lower AND unit applies a 1 to the RESET terminal of the flip-flop, causing y to change to 0.
50. Type-T, type-D, and R-S operations can be accomplished with the J-K flip-flop.
51. The DIRECT RESET input is used to clear all units in a memory to the same initial conditions.
52. The type-D operation is accomplished by placing the J and K terminals at opposite input states, "inhibiting" one input transistor and actively connecting the other to the circuit.

NOTICE: To keep our lessons up-to-date, we add material to the texts from time to time, and make occasional changes in the examination questions. Therefore, when checking your answers with this check sheet, look at the code LETTER (A, B, C, etc.) on your examination sheet to determine which of the following groups of answers apply to the lesson which you have.

5240B

1. C - NOR AND NAND CIRCUITS -- FORM THE BASIS FOR MOST DIGITAL IC's.

Most discrete logic systems rely heavily on the use of combinations of the AND and OR functions. NOT functions are also used, but as little as possible due to the additional cost. The minimization of numbers of components in discrete systems is especially important when low cost is a must. Since computers may require hundreds of thousands of functions, a savings of only a few cents per circuit becomes critical. The cost factor is not of major importance in the manufacture of IC's. All components are "manufactured" simultaneously in a step-by-step sequence which must be followed, regardless of the number of components. This step-by-step "manufacturing" sequence continues as long as there is enough physical space on the IC chip.

The NOR and NAND functions are based on using the CE amplifier arrangement as the NOT function. It takes little effort or cost to add this additional stage. Therefore, IC's are extensively utilized in all modern digital computers.

2. B - A NEGATIVE LOGIC ASSIGNMENT FOR AN IC -- PRODUCES THE NAND FUNCTION IF THE CIRCUIT IS POSITIVE - LOGIC NOR.

Reversing the logic assignments for a given circuit results in the creation of the "dual" of the circuit function. Reversing the logic assignments for the positive logic NAND circuit results in the negative logic NOR circuit.

3. C - TRANSISTOR-TRANSISTOR LOGIC -- HAS HIGH FAN-IN, FAN-OUT, NOISE IMMUNITY AND SWITCHING SPEED.

Transistor-transistor logic (TTL) provides high noise immunity, a large number of inputs and outputs, and high switching speeds. In effect, each of the emitter elements on the input transistor acts as a separate emitter-base diode, all feeding into the same active amplifying device.

4. C - WITH RESPECT TO SATURATED TYPE LOGIC, NONSATURATED LOGIC -- PROVIDES FASTER SWITCHING AND LOWER NOISE IMMUNITY.

The complete elimination of the storage time effect in switching can only be obtained by not driving the transistors into the saturation mode. Driving the transistors only in the nonsaturated mode enables the high and low voltage states to be standard fixed values rather than near-zero and saturation. The voltage swing is then small and the switching speed is considerably enhanced. The most undesirable effect makes nonsaturated operation more sensitive to erroneous switching than any other logic families.

5. A - FAN-IN AND FAN-OUT FOR AN IC -- EXPRESS INPUT AND OUTPUT LOADING CAPABILITIES.

Fan-in is defined as the number of independent inputs to a logic unit. Fan-out is defined as the capability of a logical unit to drive (in parallel) a specific number of equivalent load elements by one logical output terminal. It is standard practice to rate an IC input or output terminal in terms of a load factor. The input and output load factor must be compatible.

6. D - HIGH THRESHOLD LOGIC IS USED -- WHEN A HIGH NOISE MARGIN IS DESIRED.

When extremely high noise immunity is required without especially high-speed operation, saturated switching circuits called high threshold logic are used.

7. A - A TYPE D FLIP-FLOP IC -- HOLDS THE LOGIC VALUE AT TERMINAL A FOR THE DURATION OF THE EXISTING GATE PULSE.

The type D flip-flop is a common type of gated arrangement. In this form, the application of the gate voltage delays the logic value at terminal A, holding it for the duration of the existing gate pulse.

8. A - THE R-S FLIP-FLOP IC -- MUST HAVE A LOGICAL 1 INPUT AT THE SET TERMINAL IF ITS OUTPUT IS TO CHANGE WHEN $\overline{Y} = 1$.

When a logical 1 is applied to the S input of a R-S flip-flop, the flip-flop output at Y goes to a logical 1 and the output at $\overline{Y}$ goes to a logical 0.

9. B - A TYPE-T FLIP-FLOP -- CHANGES ONLY ON ALTERNATE STATES.

With the type T flip-flop the inverse of the output is gated to the input, but is delayed by one gating interval. With this arrangement the output state of the flip-flop changes each time a particular gating signal condition occurs. The effect is to produce an output whose frequency is half of the gating frequency.

5240B (CONT'D)

10. D - A J-K FLIP-FLOP IC -- ALWAYS CHANGES STATES WHEN BOTH INPUTS ARE LOGICAL 1 SIMULTANEOUSLY.

Operating independently, a logical 1 at either the J or K inputs causes the flip-flop to switch states. With a logical 1 at both J and K simultaneously (usually achieved by tying J and K together), the circuit operates as a type-T flip-flop. With the J and K terminals always at opposite states (achieved by placing an inverter between J and K), and a control signal on the G input, the circuit produces type D operation.

5240A

All explanations are the same as for 5240B except for that given below.

10. D - A J-K FLIP-FLOP IC -- DOES NOT HAVE ANY INDETERMINATE INPUT CONDITIONS.

Operating independently, a logical 1 at either the J or K inputs causes the flip-flop to switch states. With a logical 1 at both J and K simultaneously (usually achieved by tying J and K together), the circuit operates as a type-T flip-flop. With the J and K terminals always at opposite states (achieved by placing an inverter between J and K), and a control signal on the G input, the circuit produces type D operation.

QUESTIONS

IMPORTANT—These instructions **MUST** be accurately followed to avoid loss, or errors in grading.

Indicate your answer on this sheet by filling in the box for the most correct answer to each question, as illustrated in the example below.

When all questions have been answered, place the answer card in the proper position to line up the boxes on the card with the boxes on the sheet.

Next, copy the complete lesson code into the space provided on the card, and fill in the answer boxes to correspond with those previously filled in on this sheet.

Before mailing, be certain your correct student number, name and address appear on the card.

LESSON CODE
5240B

Example: The month of March consists of

- | | | |
|---|-------------------------------------|---------------|
| A | ☐ | (A) 365 days. |
| B | ☒ | (B) 31 days. |
| C | ☐ | (C) 6 hours. |
| D | ☐ | (D) 9 months. |

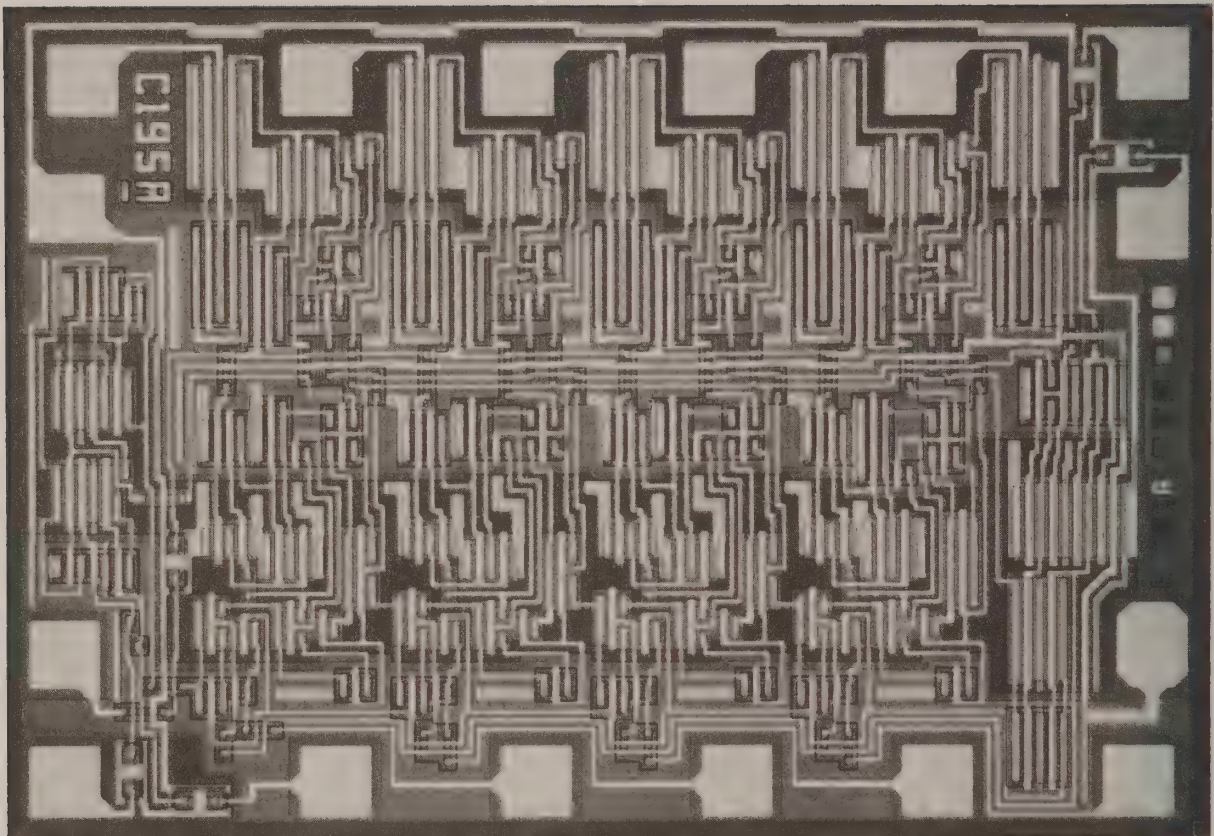
1. A ☐ NOR and NAND circuits
 B ☐ (A) are seldom used in IC's. (B) use common collector transistor arrangements. (C) form the basis
 C ☒ for most digital IC's. (D) have a very limited fan-in capability.
 D ☐
2. A ☒ A negative logic assignment for an IC
 B ☐ (A) produces the dual of the positive logic assignment, which is the same as the complementary function.
 C ☐ (B) produces the NAND function if the circuit is positive-logic NOR. (C) produces the complement
 D ☐ of the positive logic assignment. (D) has no effect on the logic operation.
3. A ☐ Transistor-transistor logic
 B ☐ (A) has high noise immunity and switching speed and a low fan-in and fan-out. (B) requires the use of
 C ☒ both NPN and PNP transistors. (C) has high fan-in, fan-out, noise immunity and switching speed.
 D ☐ (D) has large fan-in and fan-out and low noise immunity.
4. A ☐ With respect to saturated type logic, nonsaturated logic
 B ☐ (A) only provides faster switching. (B) only provides higher noise immunity. (C) provides faster switch-
 C ☒ ing and lower noise immunity. (D) offers no operating advantages.
 D ☐
5. A ☒ Fan-in and fan-out for an IC
 B ☐ (A) express input and output loading capabilities. (B) do not change from one logic type to another.
 C ☐ (C) are always equal to one another. (D) cannot be altered.
 D ☐ High threshold logic is used
6. A ☐ (A) when the fastest possible switching is essential. (B) when fast switching and low power dissipation
 B ☐ are desired. (C) when extremely low power dissipation is essential. (D) when a high noise margin is
 C ☐ desired.
 D ☒ A type-D flip-flop IC
7. A ☒ (A) holds the logic value of terminal A for the duration of the existing gate pulse. (B) can only operate
 B ☐ in an asynchronous mode. (C) operates in a manner identical to the R-S flip-flop. (D) changes output
 C ☐ states each time the information input signal changes states.
 D ☐ The R-S flip-flop IC
8. A ☒ (A) must have a logical 1 input at the SET terminal if its output is to change when $\bar{y} = 1$. (B) must have
 B ☐ a pulse at the RESET terminal if it is to change states when $y = 0$. (C) has two sets of indeterminate
 C ☐ input conditions. (D) must have a pulse at the SET terminal if it is to change states when $\bar{y} = 0$.
 D ☐ A type-T flip-flop IC
9. A ☐ (A) output changes states each time the input signal changes states. (B) changes only on alternate states.
 B ☒ (C) output always changes states when the input transitions from a high to a low value. (D) can be used
 C ☐ to divide the input frequency by any desired integer value.
 D ☐
10. A ☐ A J-K flip-flop IC
 B ☐ (A) can only be operated in a synchronous mode. (B) only changes output states each time the input
 C ☐ state changes. (C) has an indeterminate output that is the inverse of the indeterminate R-S flip-flop
 D ☒ condition. (D) always changes states when both inputs are logical 1 simultaneously.

REGISTER AND COUNTER CIRCUITS

5245

© COPYRIGHT 1973
BELL & HOWELL SCHOOLS, INC.
ALL RIGHTS RESERVED





This photograph shows the integrated circuit chip of a CMOS four-bit shift register. Designed as a direct replacement for a TTL shift register, the CMOS circuit features serial or parallel input, output, and shifting, with a power consumption of only 1/10,000 that of the TTL device.

Courtesy National Semiconductor

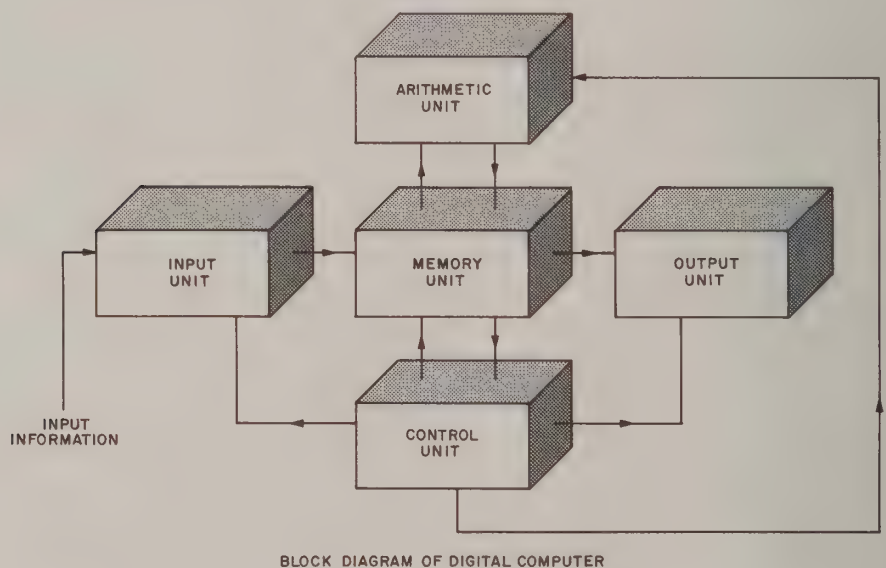
CONTENTS

Digital System Timing	Page 5
Timing Static Logic Systems	Page 8
Timing Dynamic Logic Systems	Page 9
Counters	Page 11
Binary Counters	Page 12
Decade Counters	Page 16
Basic Register Operations	Page 19
Shift Registers	Page 22
Static Shift Registers	Page 23
Dynamic Shift Registers	Page 29
Parallel Addition and Subtraction	Page 33
Parallel-Serial/Serial-Parallel Conversions	Page 35
Multiplication Logic	Page 36
Division Logic	Page 37

*Man is a special being, and if left to himself, in an
isolated condition, would be one of the weakest
creatures; but associated with his kind, he works
wonders.
—Daniel Webster*

REGISTER AND COUNTER CIRCUITS

A basic digital computer system is shown in block diagram form in Figure 1. Such systems are very flexible since the set of logical functions they perform can be changed by inserting a new **PROGRAM**. This is accomplished through a control unit which directs the sequence of operations for all other units. The program (or any other pertinent information) is entered through the input unit, then transferred to the internal memory unit. As time progresses, the control unit sequences through the set of instructions, initiating whatever other actions are required. These actions may include the operation of the arithmetic unit (which performs additions, subtractions, multiplications, etc.); the storage of new data in the memory unit; or the activation of an output unit, which provides the results for human interpretation (CRT display, type print-out, etc.).



BLOCK DIAGRAM OF DIGITAL COMPUTER

Figure 1

The basic logic functions (AND, OR, NOT), simple combinations of these functions (NAND, NOR, INHIBIT, EXCLUSIVE-OR, etc.), and any other desired combinational functions form a fundamental portion of every digital system. Flip-flops also form an important fundamental portion of every digital system. They are used as **COUNTERS** and temporary storage devices. In some cases, flip-flops perform arithmetic operations. In the last two applications, the combinations of flip-flops are known as **REGISTERS**.

Timing is very important in digital systems. Every transfer of information, arithmetic operation, or storage of information must occur at a precise time and fit into a carefully detailed schedule. Pulse generators in the digital system develop the timing pulses used to synchronize, or time, the various

circuits in the system. These timing pulses are usually called clock pulses. The pulse generator which develops the timing pulses is usually called the clock generator or simply the **CLOCK**.

In this lesson we examine digital system timing and the operation of digital counters and registers. The counters and registers are presented in terms of their "functional" operation, since the details of flip-flop (bistable multivibrator) circuit operation have already been discussed. In modern digital systems, you are likely to encounter IC counters and registers, rather than discrete circuits. Therefore, you will be more concerned with functional performance than with circuit component performance.

DIGITAL SYSTEM TIMING

Information in a digital system is processed in the form of some binary code (BCD, Gray, etc.). Groups of bits (a code group) represent a portion of each character or a numerical digit in the information being processed. It has become common practice to refer to each group of pulses as a "digit." This is a rather "loose" use of the term, since it just refers to a group of pulses which may or may not actually represent a numerical digit.

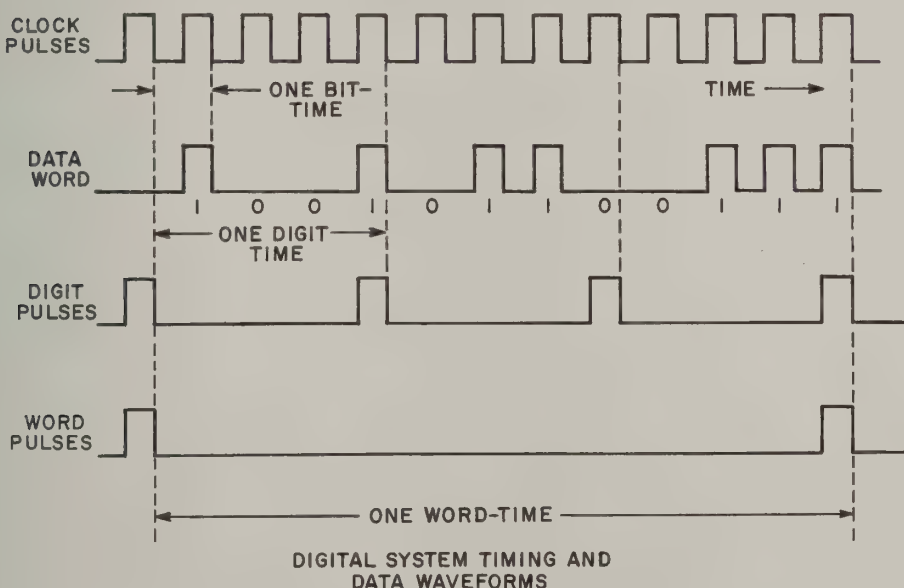


Figure 2

Pulses used in timing or synchronizing information in a typical digital system are shown in Figure 2. Clock pulses are shown on the top line. They consist of a number of equally spaced rectangular pulses. In many digital systems, the clock pulses are generated by a crystal controlled oscillator.

The output of this oscillator is applied to a waveshaping circuit, where it is converted to a series of rectangular pulses or a square wave. In some systems the output of a sinewave oscillator is used for the clock pulses. In other systems the clock pulses are narrow pulses or spikes of short duration, generated by a pulse generator, such as a multivibrator differentiator combination.

The second line in Figure 2 shows a group of digits making up a **DATA WORD**. In this example, there are three digits in the data word and each digit is composed of a group of four bits. The number of digits in a data word (and the number of bits in a digit) varies from system to system. The word, processed as a composite unit, is the basic block of information in a digital system. One word may be an instruction (telling the system to add, subtract, take data from storage, etc.), a number, alphanumeric character, or a combination of these.

Notice the relationship between the clock pulses and the presence or absence of pulses in the data word. Each bit in the data word occurs at the same time as does a clock pulse. The **BIT TIME** is the time between the trailing edge of one clock pulse and the trailing edge of the next clock pulse. Each bit in the data word occurs during one bit time. The one bit time is the smallest unit of time recognized by the system.

To separate the digits in a word, and to separate one word from another, additional timing signals (derived from the clock pulses) are needed. The individual digits in a word are separated by digit pulses. The digit pulses are shown on line three in Figure 2. All the pulses in the data word between the trailing edge of one digit pulse and the trailing edge of the next pulse make up one digit. The time between the trailing edges of the digit pulses is called the **DIGIT TIME**. The word pulses are shown on line four in Figure 2. All the pulses in the data word, between the trailing edge of one word pulse and the trailing edge of the next word pulse, make up one word. The time between the trailing edges of the word pulses is called the **WORD TIME**.

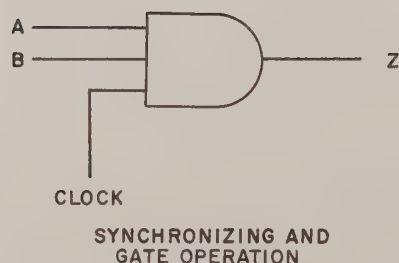


Figure
3

The timing signals shown in Figure 2 are typical of the types used in digital systems; however, there are variations among systems. For example, in one system the digit time may be the time between the leading edges of consecutive digit pulses. In another system, the digit time may be the time between the trailing edges of consecutive digit pulses.

To synchronize the operation of the various circuits in a digital system, the circuits are arranged so they operate only when a clock pulse is applied. Figure 3 shows an arrangement for synchronizing the operation of an AND gate. The clock pulse acts as another input signal. Both inputs A and B, as

well as the clock signal input, must be 1 before a 1 output signal at Z is developed. This action is illustrated in Figure 4. Notice that an output at Z occurs only when both the inputs at A and B, as well as a clock pulse, are at logic 1. The clock signal is considered to be 1 when present and 0 when not present. If the clock pulses are lost, no output signal is developed by the AND gate, even though both inputs at A and B are 1.

In Figure 4 a logical operation is performed during each bit time. The AND gate can develop an output signal only during the time the clock pulse is present. The duration of each clock pulse is the time during which the circuit performs its logical operation or decision. For this reason the time duration of the clock pulse is sometimes called the **DECISION TIME**. This is usually one-half the bit time.

In Figure 4, input signals A and B are already synchronized with the clock pulses, and synchronizing the operation of the AND gate seems unnecessary. In the actual system, the signals would never be exactly synchronized with the clock pulses. This is due to unavoidable delays in the system. The A and B signals would therefore not be in exact synchronization with the clock pulses when they are applied to the AND gate. Normally, it is not necessary to synchronize each logic circuit in the system. The circuits are synchronized only at the points in the system where the timing may change enough to cause an error.

An OR gate is synchronized in a somewhat different manner. Before the signals are applied to an OR gate, each one is passed through a separate two-input AND gate and synchronized with the clock pulses. The signals are then exactly synchronized with the clock pulses before they are applied to the OR gate. Applying clock pulses to one of the OR gate inputs would not synchronize the operation of the OR gate. Remember, an OR gate develops a 1 output signal when any or all the input signals are 1.

Besides synchronizing the logic circuits, it is also common to synchronize the input and output units of a digital system. Figure 5 shows a block diagram of a typical input unit arrangement. The input unit may be a paper tape reader, a punched card reader, a keyboard, or any other type of input unit. The 0 and 1 signals from the input unit are stored in a **SHIFT REGISTER**. A shift register consists of a series of flip-flops connected to form a storage unit. Each flip-flop in the shift register stores one bit of information. The shift register contains enough flip-flops to hold the word size the system is designed to handle. When all the bits for a single word have been applied to the shift register from the input unit, clock pulses are applied to the shift register. They transfer the bits stored in the shift register into the system, one bit at a time. Each bit of information fed into the system is exactly synchronized with the clock pulses.

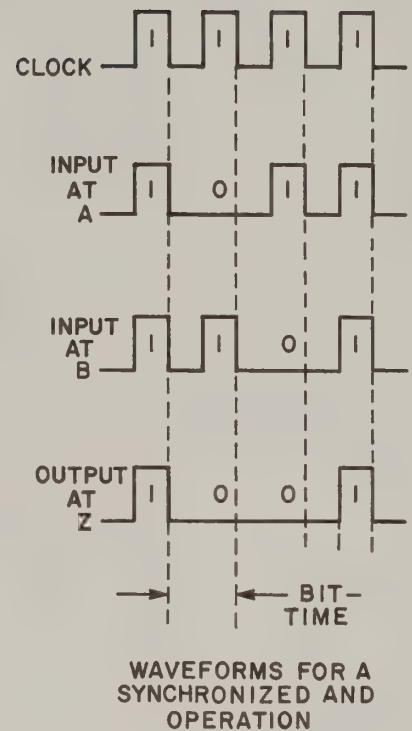


Figure
4

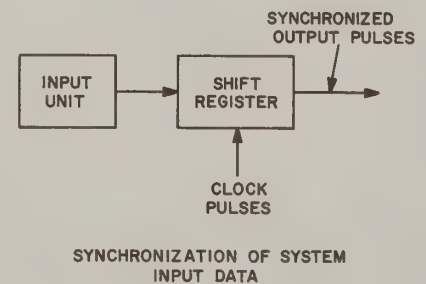


Figure
5

With the arrangement shown in Figure 5, the input information is synchronized with the clock pulses. As this information is processed by the system, it is resynchronized by applying the clock pulses to the logic circuits (as shown in Figure 3).

Most digital systems can process information at a much greater rate than the input unit can feed it in. To overcome this speed mismatch, the input information is usually placed in a storage unit and then applied to the system at a rate the system is designed to handle. The arrangement of Figure 5 performs this function by synchronizing the input information with the clock pulses. The input unit can place information into the shift register at any desired rate (or at random, if necessary). Once the input information is stored in the shift register, it can be shifted out of the register at a rate determined by the pulse repetition frequency of the clock pulses and system design specifications. When a shift register is used as a speed-matching device, as in Figure 5, it is usually called a **BUFFER**.

Placing information into a shift register at one rate and shifting it out at another rate can be done whenever changing the rate of flow in a digital system is desirable. Inside a digital computer, this arrangement can be used to match the speed of the arithmetic unit to that of the memory unit. Normally, an arithmetic unit can process information much faster than a memory unit can read-in or read-out information. The same situation occurs in a digital control system where a high-speed digital computer may be controlling a very slow metal-working machine. The output signals from the computer must be stored and then applied to the machine as they are needed.

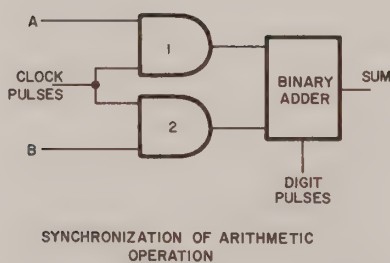


Figure
6

Another example of digital system synchronization is shown in Figure 6. Two BCD numbers are added together in this circuit. The BCD bits are synchronized with the clock pulses in AND gates 1 and 2 before being applied to the adder circuit. This assures that the bits in each number are accurately timed to the clock pulses. Digital words, each consisting of a number of digits, are inputted at A and B. In the BCD code, every digit is represented by a group of four bits. To distinguish one group of bits from another in the adder, digit pulses are applied. Digit pulses separate one digit (group of four bits) from another (as shown in Figure 2). This arrangement is necessary since an incorrect sum will result if the bits for each number are intermixed.

Timing Static Logic Systems

There are two types of logic systems used in digital systems. These are dc or **STATIC LOGIC SYSTEMS** and ac or **PULSE LOGIC SYSTEMS**. In static logic systems, the 0 and 1 signals are dc levels. In pulse logic systems, the 0 and 1 signals are pulses. The timing or synchronization in each type of logic system is somewhat different.

In static logic systems, the signal voltages representing 0 and 1 cannot reach their maximum values instantaneously. Circuit capacitance and storage effects in the semiconductors used to construct the logic circuits prevent instantaneous changes in signal voltages. For this reason, the signals appearing at the input to a logic circuit may look like the inputs at A and B in Figure 7. The positive signal level represents 1 and the zero signal level, 0. The clock pulses in a static logic system are usually rectangular pulses with very sharp leading and trailing edges, so an accurate synchronization can be obtained.

Since the signal voltages do not reach their maximum values instantaneously, a logic circuit may produce an incorrect output during the time the signals are changing. This action may be eliminated by postponing the logical operation until the signals have had time to reach their steady-state values. This may be readily accomplished by timing the clock pulses so they occur after the signals have reached their steady-state levels. Assume that inputs A and B, along with the clock pulses, are applied to a three-input AND gate. The AND gate develops a 1 output signal only when input signals at A and B and the clock pulses are 1. With this arrangement, the output signal at Z appears as shown in Figure 7.

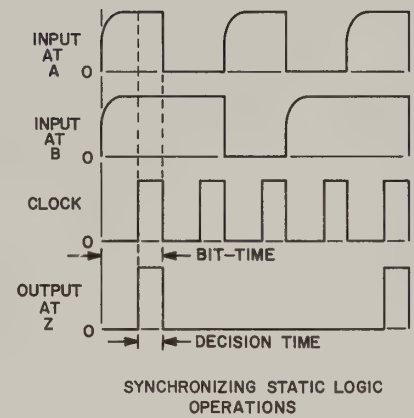


Figure
7

In Figure 7 the output occurs only after the input waveforms have reached their steady state. The logic operation is performed only during the second half of the bit time (while the clock pulse is present). The decision time for a static system may be very short, giving a pulse-like nature to the output of each logical device.

The output pulses are used to trigger multivibrators used to maintain the longer duration static waveforms.

Timing Dynamic Logic Systems

In pulse or **DYNAMIC LOGIC SYSTEMS**, the 0 and 1 signals are short duration pulses. In one system a 1 may be represented by a positive pulse and a 0 by the absence of a pulse. A 1 can also be represented by a negative pulse and a 0 by the absence of a pulse. A 1 can also be represented by a positive pulse and a 0 represented by a negative pulse, or vice versa.

Most memory units use pulse logic. The devices used as memory units include magnetic cores, magnetic drums, delay lines, etc. All use pulses of voltage or current to read-in and read-out information.

The pulses in a dynamic logic system also cannot reach their maximum values instantaneously due to the circuit capacitance and storage delays in

REGISTER AND COUNTER CIRCUITS

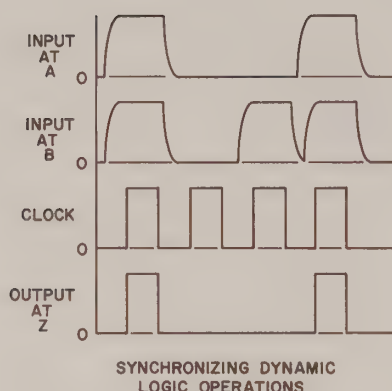


Figure 8

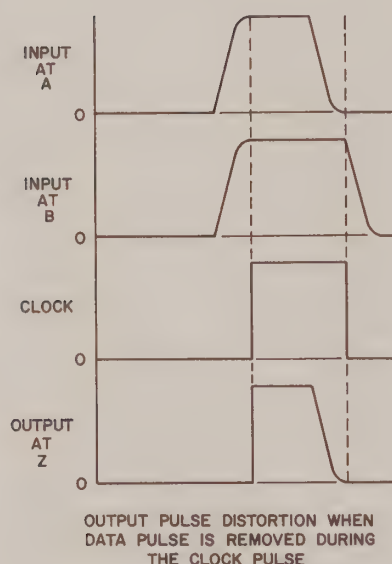


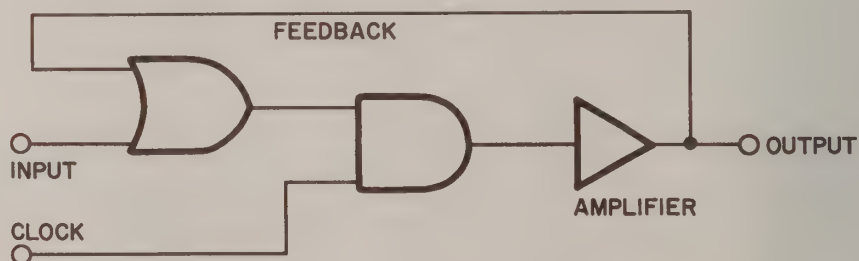
Figure 9

the semiconductors used in the circuit. The signals applied to an AND gate in a dynamic logic system might appear as shown in Figure 8.

An error or an incorrect output from a gate circuit can also occur in pulse logic if the logic circuit performs its logical operation while the input signals are changing. To prevent this, the circuit is arranged so it performs its logical operation only when the clock pulse is present. The clock pulse is timed so it occurs after the signals have reached their proper levels. Applying inputs at A and B along with the clock pulses of Figure 8 to an AND gate produces the output at Z, as shown.

Notice that the input signals of Figure 8 do not have the same time duration, nor do they have the same timing. This is a result of unavoidable delays in the system. However, timing the clock pulses so they occur after the signals have completely changed assures that the output signals from the logic circuits are correct and accurately timed.

If either of the AND gate input signals of Figure 8 changes while the clock pulse is applied, the time duration of the output pulse is affected, as shown in Figure 9. Notice that the input at A decreases to zero before the input at B and the clock pulse decrease to zero. This reduces the time duration of the output pulse to something less than the clock pulse. To overcome this problem, the output signal of the gate may be applied to a **RETIMING AND RESHAPING AMPLIFIER**. This type of amplifier is triggered by the leading edge of the input signal and delivers an output pulse whose time duration and waveshape is independent of the input signal. The time duration of the output signal from the retiming and reshaping amplifier is equal to the time duration of the clock pulses, whether the input signal time duration is longer or shorter than that of the clock pulse.



LOGIC DIAGRAM FOR RETIMING AND RESHAPING AMPLIFIER

Figure 10

A logic diagram of a typical retiming and reshaping amplifier is shown in Figure 10. This circuit consists of a pulse amplifier (represented by the triangular symbol) and two gate circuits: an AND gate and an OR gate. The

output of the amplifier is fed back to the OR gate along with the input signal. The OR gate output and the clock pulses are applied to the AND gate. The output of the AND gate is then applied to the input of the pulse amplifier. The amplifier circuit of Figure 10 is sometimes called a pulse-regenerative amplifier, due to the feedback provided in the circuit.

With no input signal applied, no output signal is developed by the OR gate. As a result, the only input to the AND gate at this time are the clock pulses. Thus, no signal is delivered to the pulse amplifier, and no output signal is developed. Now, assume that an input signal is applied to the OR gate. The OR gate applies an input signal to the AND gate. With the clock pulse and the signal from the OR gate both present, the AND gate delivers a signal to the pulse amplifier. The output signal from the pulse amplifier is applied back to the OR gate, where it combines with the input signal. The duration or width of the output pulse is determined by the width of the clock pulse. As soon as the clock pulse decreases to zero, the AND gate output signal no longer appears and the input to the pulse amplifier is removed. Thus, the time duration of the output signal is equal to the clock pulse. This is true even when the time duration of the input signal is longer than the clock pulse. As soon as the clock pulse decreases to zero, no signal is delivered to the amplifier.

Assume that the time duration of the input signal is less than that of the clock pulse. At the instant the input pulse has decreased to zero, some feedback signal remains and the OR gate still delivers a signal to the AND gate. This signal, together with the clock pulse, develops an input signal for the pulse amplifier from the AND gate. The clock pulse, together with the feedback signal from the OR gate, develops an input signal for the pulse amplifier from the AND gate. This input signal remains as long as the clock pulse remains. In turn, the pulse amplifier produces a feedback signal as well as an output signal. Thus, the time duration of the output signal from the retiming and reshaping amplifier is still fixed by the clock pulse time duration. The shape of the output pulse from the retiming and reshaping amplifier is determined by the characteristics of the pulse amplifier.

COUNTERS

Electronic counters are often the basic units in digital measurement and control systems. In radiation work, counters are used to count subatomic particles. Counters are also used to calibrate oscillators by counting the number of cycles produced per second at the oscillator output. In larger digital systems (such as computers) counters are often used for "book-keeping" within the system; they keep track of things. For example, a counter might count the number of shifts performed by a shift register, the number of instructions selected from a recorded program, or the number of pulses of a known frequency to provide a measure of time.

An electronic counter keeps track of the number of pulses applied to its input during a given time. These pulses are often supplied by the system clock. The cumulative count output of the counter is normally displayed on a control panel.

Binary Counters

When flip-flops are connected together in cascade, they form counters. A four-stage **BINARY COUNTER** using type-T flip-flops with a **DIRECT RESET** capability is shown in Figure 11. For text consistency, all flip-flop outputs have been labeled with lower case letters (y or $\bar{y}$). Since manufacturers of flip-flops may use all upper case letters, or all lower case letters, or a combination of upper and lower case letters to identify inputs and outputs, the figures in this lesson may contain any of their labeling styles. Note that in Figure 11 and in many following figures the signal flow is from right to left. This permits a direct read-out of the flip-flop states. The $\bar{y}$ output of each stage is connected to the trigger input of the next stage. The y output of each stage indicates the "count." The "count" may be displayed utilizing a neon lamp for each stage (lamp lit for $y = 1$, extinguished for $y = 0$). Other forms of output display can also be used. Each time a new count is desired, a pulse is applied to R_D , thus clearing the output (setting the y outputs to 0). Four pulses are shown at the counter input which is to be counted, and a pulse is about to be applied to the **DIRECT RESET** line. In Figure 12A, a pulse is applied to the trigger input of the first (right-most) stage causing FF_1 to assume the $y_1 = 1$ state; the count becomes 0001 (decimal 1). This leaves three pulses which are to be applied to FF_1 . The count can be read directly from the figure since the y outputs of FF_4 , FF_3 and FF_2 are 0 while the y output of FF_1 is 1 at the end of the first input pulse.

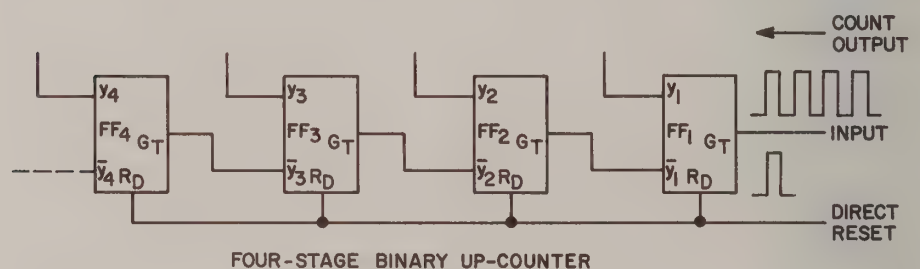


Figure 11

This count is retained as long as no other pulses are applied, as long as R_D is not pulsed, or until the circuit is de-energized. In Figure 12B, a second pulse is applied. This pulse switches the first stage (y_1) to 0 ($\bar{y}_1 = 1$), leaving two pulses to be applied to FF_1 . The pulse from $\bar{y}_1$ causes the second flip-flop to assume the $y_2 = 1$ state, and the count becomes 0010 (decimal 2). In Figure 12C, a third pulse is applied. This pulse switches the first stage (y_1) to 1 and leaves the second stage in the $y_2 = 1$ state, and the count becomes 0011 (decimal 3). In Figure 12D, the last pulse is applied, causing y_1 to switch to 0. The pulse at $\bar{y}_1$ causes the second stage (FF_2) to switch

The following Practice Exercise questions cover the subjects which you have just studied. They are:

DIGITAL SYSTEM TIMING

TIMING STATIC LOGIC SYSTEMS

TIMING DYNAMIC LOGIC SYSTEMS

1. What is the relationship between the bits in the data word and the clock pulses in Figure 2?
2. In Figure 2, the digits in the data word are separated by the (a) digit pulses, (b) word pulses.
3. In Figure 2, one data word is separated from the next by the (a) digit pulses, (b) word pulses.
4. To synchronize an OR gate, the clock pulses are applied directly to the gate, along with the information signals. True or False?
5. In a particular digital unit, the bit time is 20 nanoseconds. The decision time for this unit is (a) _____. A data word for this unit contains four digits, each composed of eight bits. The digit time for the unit is (b) _____ and the word time is (c) _____. There are (d) _____ bits per data word.
6. All parts of a digital system use the same bit time. True or False?
7. Static logic systems utilize longer term dc waveforms, as compared to the waveforms of pulse logic systems. True or False?
8. The clock signal in a static system determines the duration of each logical decision. True or False?
9. In dynamic logic systems, the desired pulse durations are maintained by using _____.

10. The width of the output pulse from the circuit in Figure 10 is controlled by the (a) amplifier characteristics, (b) width of the clock pulse.
11. How can the amplifier in Figure 10 maintain an output pulse after the input pulse stops, assuming that the clock pulse remains at a high level?

y_2 to 0 ($\bar{y}_2 = 1$). The positive-going edge of the pulse at $\bar{y}_2$ triggers the third stage (y_3) to 1, providing a count of 0100 (decimal 4). The associated timing diagram is shown in Figure 13. Note that the positive-going edge of the input to FF₁ causes FF₁ to change states. However, the actual input to FF₂ is $\bar{y}_1$ instead of y_1 , which is shown in Figure 13. When y_1 goes negative, $\bar{y}_1$ goes positive. Thus, all of the flip-flops in Figure 12 switch at the positive-going steps of their input signals.

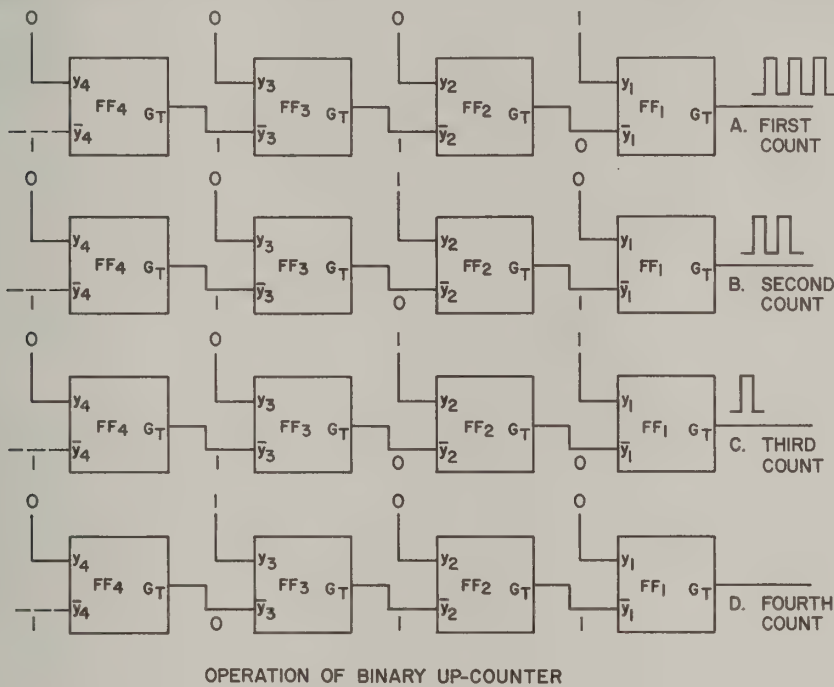


Figure 12

The complete set of digital states for the counter stages of Figure 12 is shown in Table 1. This particular arrangement is an **UP-COUNTER**. Its maximum value is limited by the number of stages. With four stages, the maximum count is 1111 (decimal 15).

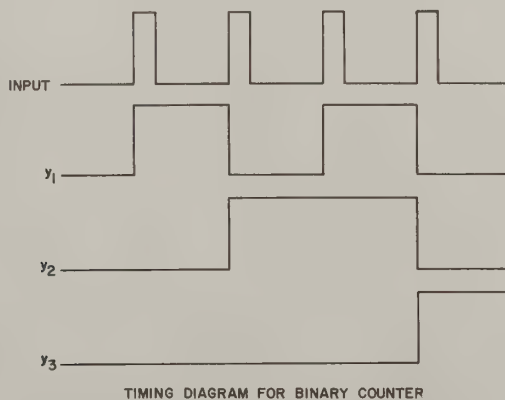


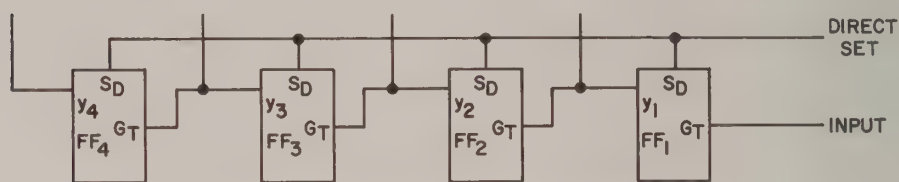
Figure 13

Count	Output State			
n	y ₁	y ₂	y ₃	y ₄
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	1	1	0	0
4	0	0	1	0
5	1	0	1	0
6	0	1	1	0
7	1	1	1	0
8	0	0	0	1
9	1	0	0	1
10	0	1	0	1
11	1	1	0	1
12	0	0	1	1
13	1	0	1	1
14	0	1	1	1
15	1	1	1	1

"Count" Data for Binary Up-Counter

Table
1

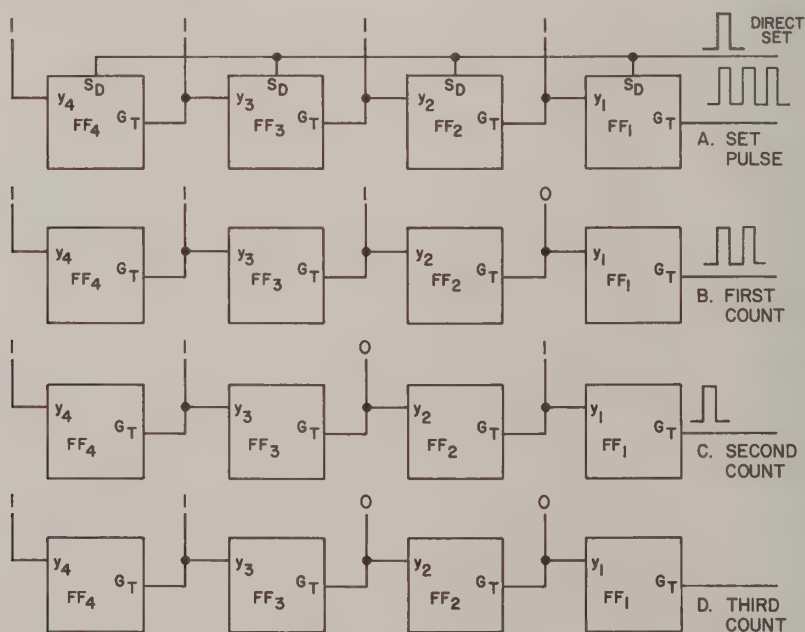
A **DOWN-COUNTER** can be formed by connecting the y output of each stage to the trigger input of the following stage. For this application, type-T flip-flops with a **DIRECT SET** capability can be used, as shown in Figure 14. Initially, all stages are set (to $y = 1$). Now, each time a pulse is applied, the counter output decreases by one. The **DIRECT SET** and **DIRECT RESET** terminals of each flip-flop stage can also be used to "start" a counter at any desired value. The count proceeds downward from this initial value.



THE BINARY DOWN-COUNTER ARRANGEMENT

Figure 14

The operation of the down-counter is shown in Figure 15. Initially, S_D in all four flip-flops (Figure 15A) is pulsed to establish the maximum count (binary 1111, or decimal 15), and there are three pulses about to enter G_T of FF_1 . The first count pulse is applied to FF_1 in Figure 15B. This causes y_1 to assume the zero state, placing the count at 1110 (decimal 14) and leav-



OPERATION OF DOWN-COUNTER

Figure 15

ing two pulses to enter the input of FF_1 . The second pulse (Figure 15C) switches y_1 to the 1 state. This then causes y_2 to go to 0. The count is then 1101 (decimal 13). The third pulse causes y_1 to return to 0; however, it does not affect the other stages (Figure 15D). Now the count is 1100 (decimal 12). The complete count cycle is shown in Table 2.

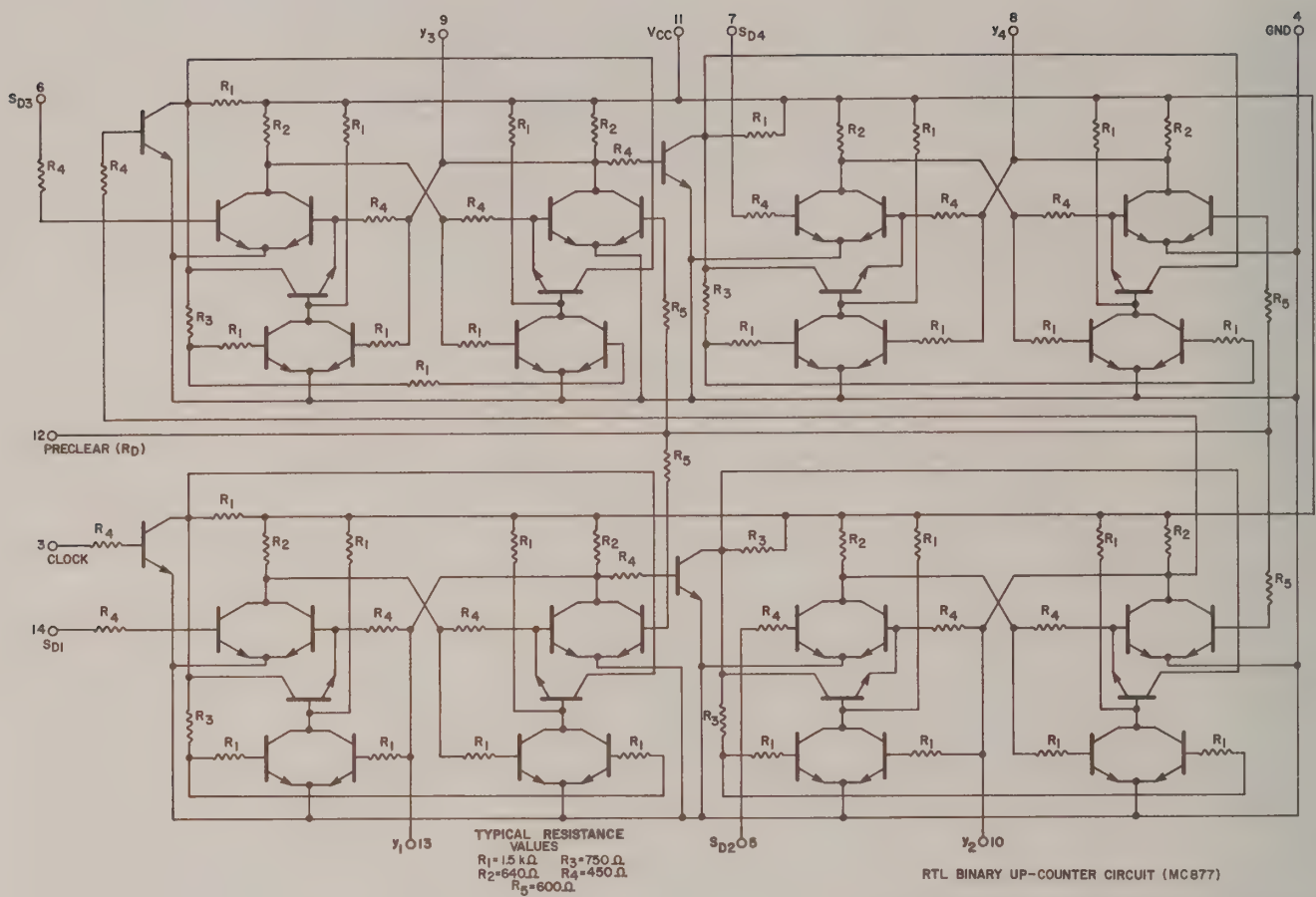
Count	Output State			
	y_1	y_2	y_3	y_4
15	1	1	1	1
14	0	1	1	1
13	1	0	1	1
12	0	0	1	1
11	1	1	0	1
10	0	1	0	1
9	1	0	0	1
8	0	0	0	1
7	1	1	1	0
6	0	1	1	0
5	1	0	1	0
4	0	0	1	0
3	1	1	0	0
2	0	1	0	0
1	1	0	0	0
0	0	0	0	0

"Count" Data for Binary Down-Counter

Table 2

An integrated circuit of an RTL binary up-counter is shown in Figure 16. This binary up-counter provides the complete up-count capability for the system in Figure 11. The IC includes a common PRECLEAR terminal (R_D) for clearing the counter. It also includes an individual DIRECT SET terminal (S_D) for each stage. This allows the user to either set the counter to any desired initial value before starting the count, or to "advance" the count to any desired value during the counting process. The maximum count of 15 associated with the IC in Figure 16 can be increased by cascading two or more similar IC's, or by selecting an IC with more stages. The maximum count increases rapidly (it is 255 if two of the four-stage IC's are used together); however, it remains in a binary form at the output.

REGISTER AND COUNTER CIRCUITS



Courtesy Motorola Semiconductor Prod. Div.

Figure 16

Decade Counters

It is also possible to arrange four flip-flop stages in the form called a **DECADE COUNTER** to count from decimal 0 to decimal 9 (1001). Another group of four flip-flops can be fed by this group, triggering them from a count of 0000 to a count of 0001 when the actual count in the first decade counter reaches decimal 10. (The original group of four goes to 0000 at this point.) Such an arrangement is known as a **DECADE REGISTER**. Each four-stage decade counter in this type of a cascade arrangement represents one of the digit-places in a decimal number, permitting operation in the binary coded decimal (BCD) mode. Such an arrangement is shown in Figure 17. Each of the vertical stacks of four units represents a single decade counter. In the figure, pulses already have been applied to the DIRECT SET terminals of each of the individual flip-flops in each decade counter in order to establish an initial count of 850,332. The count would proceed by applying pulses to the input terminal of the right-most decade.

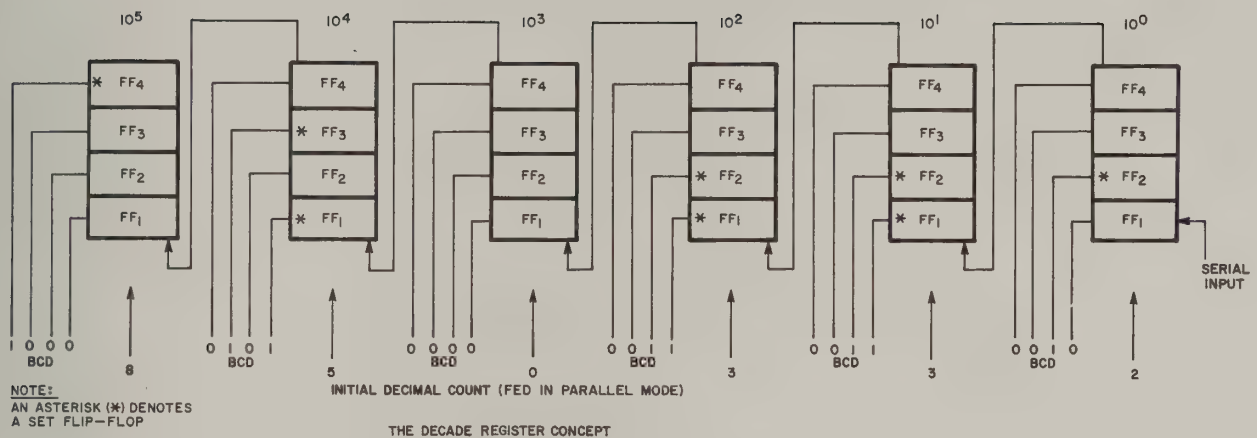


Figure 17

Figure 18 illustrates the operation of a single decade counter. Note that FF_1 is a simple divide-by-2 flip-flop and it is quite independent from the other flip-flops. FF_2 , FF_3 and FF_4 are arranged with OR gate 1 and the AND gate as a divide-by-five circuit. Other arrangements are also possible. As shown, the flip-flop cascade is basically the same as a binary up-count circuit with the $\bar{y}$ terminal of each stage applied to the gate input of a type-T circuit. The two middle stages have their y outputs as inputs to an OR gate. This OR gate, in conjunction with the y output of the last stage, feeds an AND gate. The AND gate output is tied to the DIRECT RESET terminals of the two middle flip-flops.

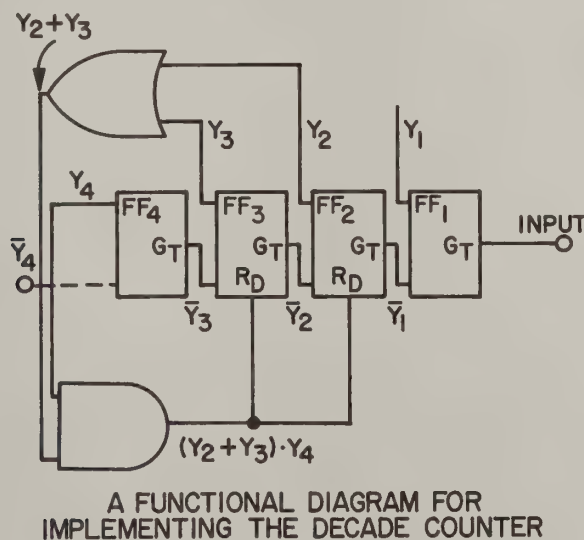
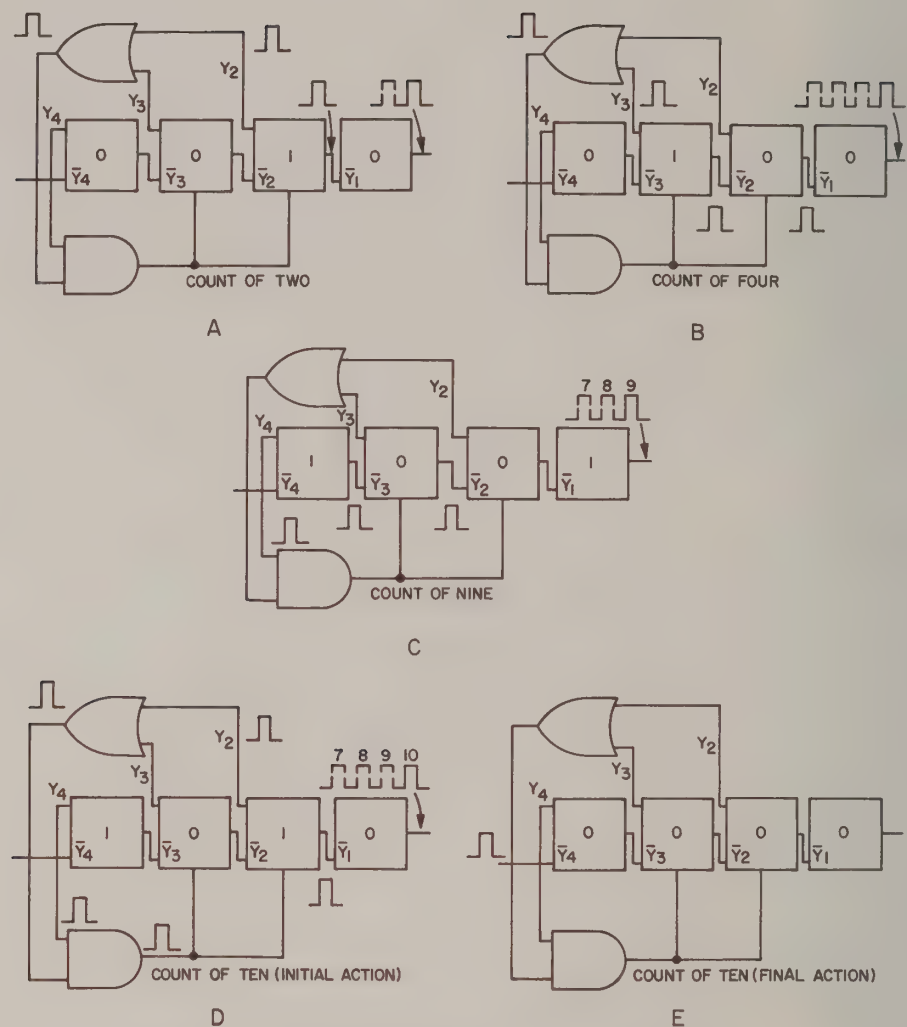


Figure 18

The OR gate and the AND gate in Figure 18 have no effect on circuit operation as the count proceeds from 0 through decimal 9.



OPERATION OF THE DECADE COUNTER ARRANGEMENT IN FIGURE 18

Figure 19

Although the OR gate puts out a pulse at the count of decimal 2 and 3 (Figure 19A) and at the count of decimal 4, 5, 6 and 7 (Figure 19B), the AND gate output remains at zero. When the count goes to decimal 9, the condition shown in Figure 19C exists. The next count (decimal 10) causes an output of 1 from the OR gate, as shown in Figure 19D. This places logical 1's on both AND gate inputs, causing a logical 1 output from the AND gate to the RESET terminals of the two middle flip-flops. This causes the two middle flip-flops to reset to zero. However, FF_2 and FF_3 are reset sequentially rather than simultaneously. FF_2 is reset first which sets FF_3 and causes another reset pulse to appear at the output of the AND gate. This second reset pulse now resets FF_3 . (When FF_2 is being reset, FF_3 is not affected; when FF_3 is being reset, FF_2 is not affected.) The reset action produces a logical 1 output from $\bar{Y}_3$, which changes the Y_4 output to zero

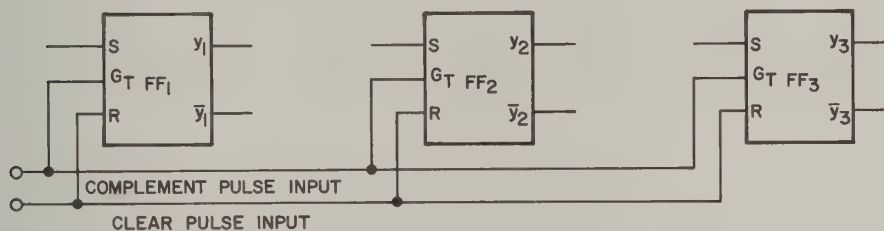
and the $\overline{Y}_4$ output to 1, which produces a count of 1 from the first stage of the next decade counter.

Now the count will continue to 20 before the AND gate can produce another logical 1 output. At the count of 20, both inputs to the AND gate are at logical 1 and the AND gate again produces a logical 1 output to reset the first decade counter to zero and drive the second decade counter to a count of decimal 2. This action continues, raising the count on the second decade counter by 1 for each count of decimal 10 on the first decade counter. At decimal 100, the OR and AND gates of the second decade counter reset the second counter to zero and initiate a count of 1 in the third decade counter (see Figure 17).

BASIC REGISTER OPERATIONS

A binary up-counter such as the one shown in Figure 11 can also serve as a register providing temporary storage for a digital computer system.

The two basic register operations are the clear-register operation and the complement operation. Figure 20 shows the logic block diagram of a three-bit storage register made up of three flip-flops which have reset, set and complement (trigger) inputs.



THE BASIC REGISTER ARRANGEMENT WITH
COMMON CLEAR AND COMPLEMENT TRIGGERS

Figure 20

A pulse at the clear pulse input removes (clears) any stored numbers by resetting all the flip-flops (to the 0 state). A pulse at the complement pulse input causes each flip-flop to switch to the opposite state. This forms the 1's complement of the original stored number. For example, if the register originally contains the number 101, after the complement pulse is applied, the register contains the number 010.

Another operation common to all registers is the transfer of data from one register to another. Figure 21 shows a transfer-after-clear circuit. In the transfer-after-clear operation, the number in register B (the lower set of flip-flops) is transferred to register A (the upper set of flip-flops). First, a

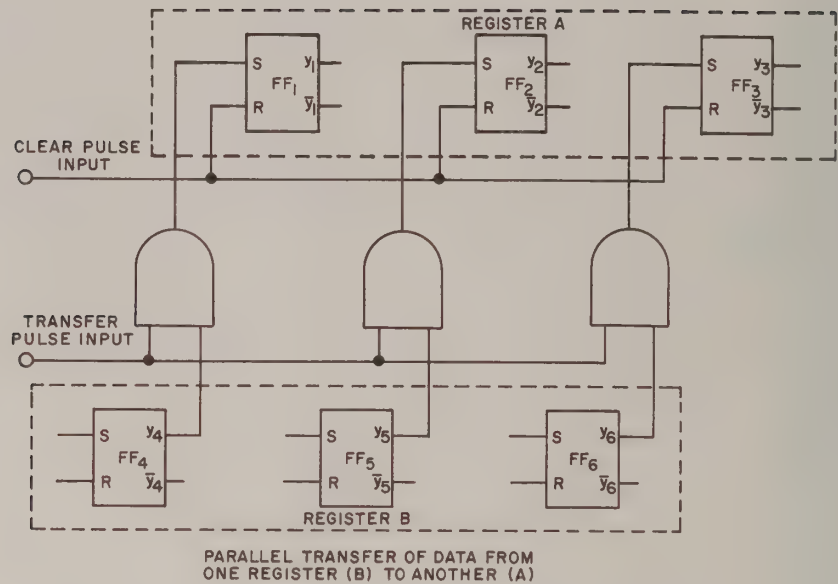


Figure 21

clear pulse clears register A. Then a pulse is applied to the transfer pulse input. This pulse is applied to one input of each transfer line AND gate. The non-inverted outputs of the register B flip-flops are applied to the other input of each AND gate. If any of the flip-flops of register B are in the 1 state (SET), the dc outputs from these flip-flops write 1's into the corresponding flip-flops in register A when the transfer pulse appears. If a flip-flop in register B contains a 0 (RESET), the corresponding AND gate does not produce an output, and the corresponding flip-flop in register A remains in the 0 state.

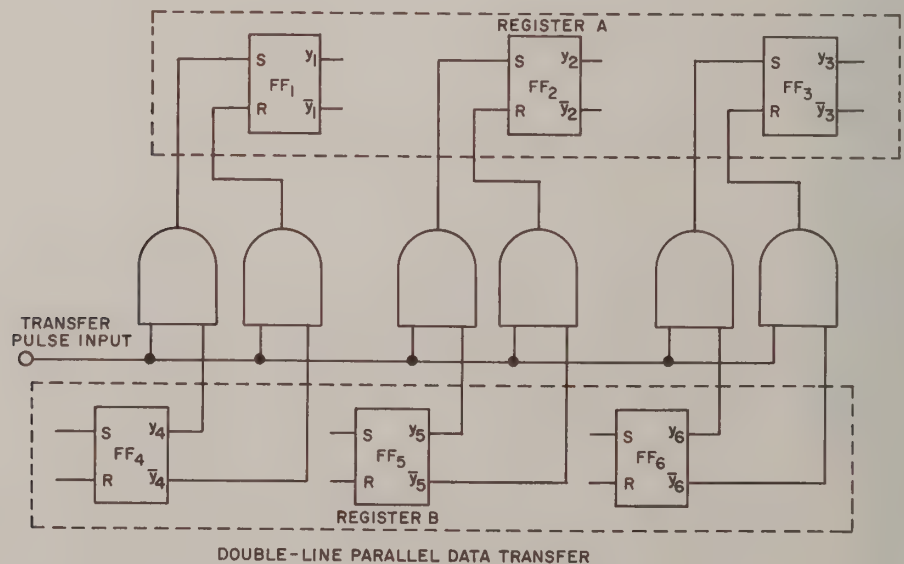


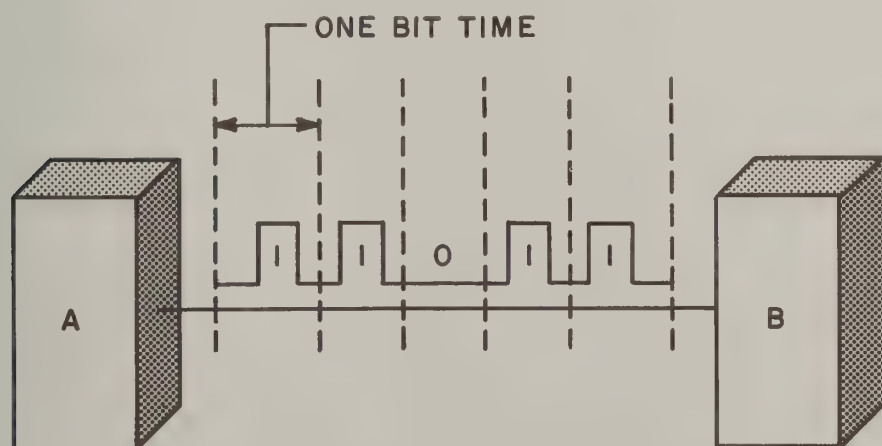
Figure 22

Figure 22 shows a double-line transfer circuit. This circuit transfers a number from register B to register A, regardless of the original states of the flip-flops in register A. A clear pulse is not required. Two AND gates are connected between each pair of corresponding flip-flops. The y output of a flip-flop in register B connects to one input of one AND gate, and the $\bar{y}$ output of the flip-flop connects to one input of the other AND gate. The outputs of the AND gates connect to the corresponding inputs in register A: y to set (S) and $\bar{y}$ to reset (R).

When the transfer pulse is applied, one of the two AND gates connected between each pair of flip-flops produces an output. These outputs switch the flip-flops in register A to the same state as the corresponding flip-flops in register B. Each flip-flop in register A receives a pulse from one of the two AND gates connected to it. If one of the flip-flops in register A contains a 0, and a pulse is applied to its reset input, the flip-flop remains in the 0 state. However, if the register A flip-flop contains a 1, a pulse applied to its reset input switches it to the 0 state.

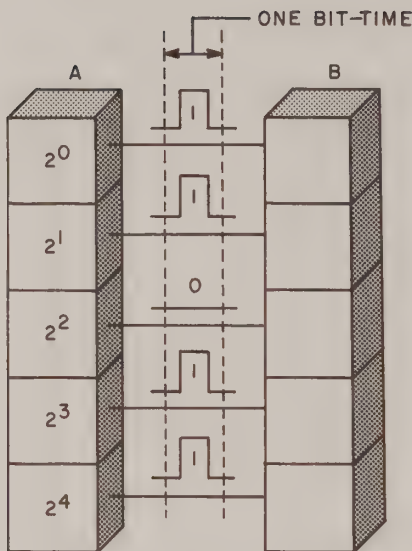
Both transfer circuits of Figures 21 and 22 provide parallel read-out and write-in capabilities. If shift registers are used in these circuits, serial input and output can also be obtained.

In a digital system, information is continually being transmitted between circuits and units in the system. Information in a digital system is coded in the form of binary digits. The two general methods of transferring information between units are called **SERIAL** and **PARALLEL TRANSMISSION**.



THE GENERAL PRINCIPLE OF
SERIAL DATA TRANSMISSION

Figure 23



THE GENERAL PRINCIPLE OF
PARALLEL DATA TRANSMISSION

Figure
24

Figure 23 shows how the five-bit binary number 11011 is transmitted serially over a single line. The number is being transmitted from Section A to Section B of a digital system. The bit on the right (closest to Section B) is the least significant bit with a weight of 2^0 or 1. The left-most bit (the last one from Section A) is the most significant bit with a weight of 2^4 or 16. In this system the presence of a pulse represents 1 and the absence of a pulse represents 0. Any of the other methods of representing bits may also be used. In this method of transmission, the binary number appears as a pulse train.

Notice that the time allowed for the representation of a bit is the same for each bit in the number. It is important to notice that in the serial transmission system of Figure 23 the second bit cannot be transmitted until the first bit has been sent; the third bit must wait until after the second has been sent; and so on for all of the bits in the number.

Another common method of transmitting binary numbers is to transmit each of the bits in a number on a separate line. This is called parallel transmission. Figure 24 shows how the five-bit binary number 11011 is transmitted over five parallel lines from Section A to Section B of a digital system. In the system shown in Figure 24, each line has a different value (or weight), depending on which bit in a number it carries. Thus, the lines having values of 1 (2^0), 2 (2^1), 8 (2^3) and 16 (2^4) all carry logic 1 pulses, while the line having a weight of 4 (2^2) carries a logic 0 pulse.

In serial transmission, only one line is needed to transmit a number (such as in Figure 23). In comparison, many transmission lines are needed for parallel transmission (such as in Figure 24). However, a complete word is transmitted in one bit time in a parallel system, and one word time in a serial system. Serial transmission has the advantage of simplicity, but parallel transmission provides faster operation. Many digital systems use serial transmission in some sections and parallel transmission in others, depending on which is the more advantageous for a particular section.

SHIFT REGISTERS

Besides storing information, **SHIFT REGISTERS** have provisions for shifting a number to the right or left with respect to the decimal point. For example, assume that a shift register is storing the binary number 101.11. If this number is shifted one place to the right, the result is 10.111. If the number is shifted one place to the left, the result is 1011.1. Notice that a shift to the right is the same as halving the binary number or dividing by 10 (decimal two). That is:

The following Practice Exercise questions cover the subjects which you have just studied. They are:

COUNTERS

BINARY COUNTERS

DECADE COUNTERS

BASIC REGISTER OPERATIONS

12. Describe the action in Figure 12 if the condition in Part D exists and a fifth pulse is applied to the input.
13. What limits the maximum value which can be accumulated by a binary up-counter?
14. What is different about the cascade connections made for an up-counter and a down-counter?
15. How can the initial count of the IC in Figure 16 be set to 0101 (decimal 5)?
16. Describe the action of the decade counter illustrated in Figure 18 when the count stands at decimal 9 and another input pulse is applied.
17. How many decade counters must be cascaded in order to count to 1,000,000?
18. If the number of counters indicated in Practice Exercise 17 were used as natural binary counters, what would the maximum count be?
19. When a register is cleared, all the flip-flops in the register are set (to the 1 state). True or False?
20. Register A of Figure 22 must be cleared before a number can be transferred from register B to register A. True or False?
21. A register holds the data word 10110011. When the complement terminal is pulsed, the data word is changed to _____.

22. In serial transmission, each bit follows the preceding one. True or False?
23. In serial transmission, the transfer of a word from one unit to another takes (a) _____ time; while, in parallel transmission, the transfer takes (b) _____ time.

Binary Division	Decimal Division
$\begin{array}{r} 10.111 \\ 10 \overline{) 101.110} \\ \underline{10} \\ 11 \\ \underline{10} \\ 11 \\ \underline{10} \\ 10 \\ \underline{10} \\ 0 \end{array}$	$\begin{array}{r} 2.875 \\ 2 \overline{) 5.75} \\ \underline{4} \\ 17 \\ \underline{16} \\ 15 \\ \underline{14} \\ 10 \\ \underline{10} \\ 0 \end{array}$

Also notice that a shift to the left is the same as doubling the binary number or multiplying by 10 (decimal two). That is:

Binary Multiplication	Decimal Multiplication
$\begin{array}{r} 101.11 \\ \times 10 \\ \hline 1011.10 \end{array}$	$\begin{array}{r} 5.75 \\ \times 2 \\ \hline 11.50 \end{array}$

This process of halving or doubling a number by shifting is often used in the arithmetic units of digital systems. One of the basic components in an arithmetic unit, called an **ACCUMULATOR**, is actually only a shift register. Shift registers are also used to convert information from serial form to parallel form or from parallel form to serial form. A serial number may be transferred into a shift register and read out in parallel, or a parallel number may be transferred into a shift register and read out in serial.

Static Shift Registers

Figure 25 shows the logic diagram of a **STATIC SHIFT REGISTER**. The register is made up of three SET-RESET flip-flops. Although the following description assumes that this is a complete three-bit register, it could be

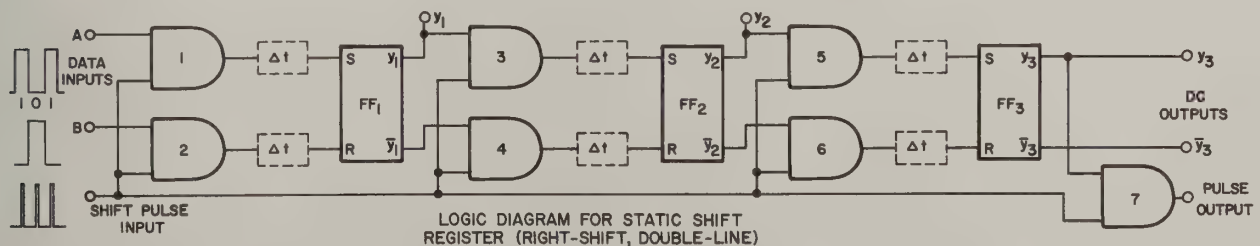


Figure 25

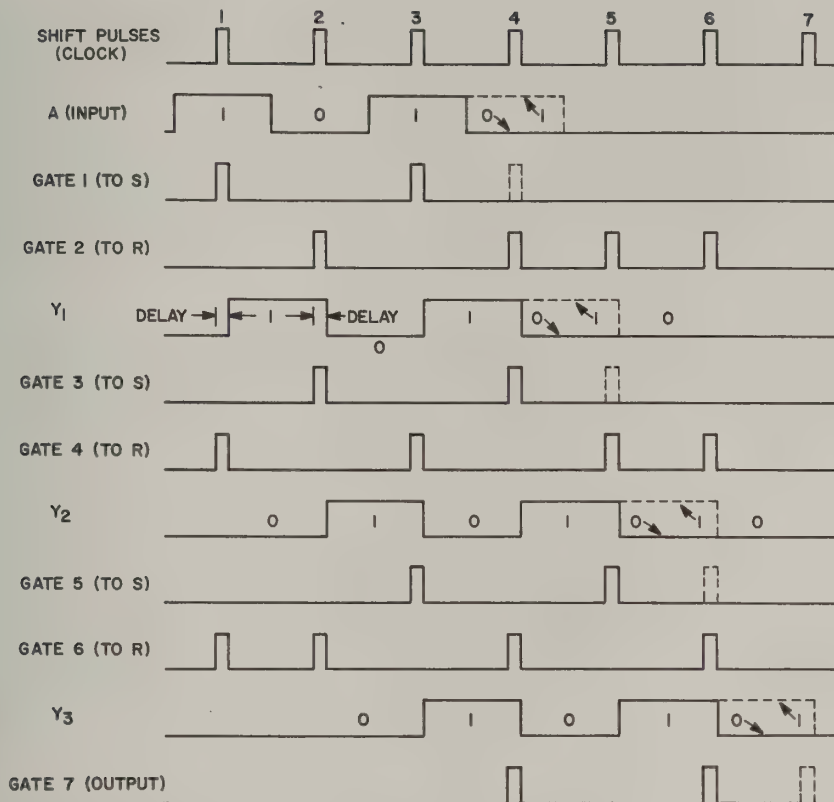
part of a larger register. In this register, the bit stored in each flip-flop is shifted one place to the right each time a shift pulse is applied. This is called a right-shift register.

The shift pulses applied to the shift pulse input of Figure 25 are actually clock pulses. Thus, the number in the register is shifted one place to the right during each bit time. Bits are inserted into FF_1 and transferred from flip-flop to flip-flop within the register by double-line transfer circuits. Since double-line transfer circuits are used, the register must have two data inputs. One input signal is the inverse of the other. The input signals may be obtained directly from the outputs of a flip-flop, or, if an inverted output is not available, the lower data input of Figure 25 may be preceded by a NOT circuit.

Before examining the operation of the register, let's review R-S flip-flops. An R-S flip-flop is set (to the 1 state) when a 1 is applied to its S (SET) input. It is reset (to the 0 state) when a 1 is applied to its R (RESET) input. When a flip-flop is in the 1 state, its y output is in the 1 state and its $\bar{y}$ output is in the 0 state. However, when a flip-flop is in the 0 state, its y output is in the 0 state and its $\bar{y}$ output is in the 1 state.

Assume that all of the flip-flops are in the 0 state and that the binary number 101 is to be shifted into the register. The waveforms are shown in Figure 26. Let us first ignore the signals that are shown with dashed lines. Shift pulses 1, 2 and 3 are for read-in and 4, 5 and 6 are for read-out. The first bit applied to the A input is a 1 (and a 0 at B). During the first shift pulse, the output pulse of AND gate 1 is in the 1 state and the output pulse of AND gate 2 is in the 0 state. Since all of the flip-flops were in the 0 state, the output pulses of AND gates 3 and 5 are in the 0 state and the output pulses of AND gates 4 and 6 are in the 1 state. The output of AND gate 1 is applied to the S input of FF_1 . This sets FF_1 . The output pulses of AND gates 4 and 6 are applied to the R inputs of FF_2 and FF_3 , respectively. FF_2 and FF_3 remain in the 0 state. Thus, after the first shift pulse, the number stored in the register is 100. The first bit at the input has been shifted into the register.

During the second shift pulse, the bit applied to the A input is a 0 (with a 1 at B). FF_1 is in the 1 state and FF_2 and FF_3 are in the 0 state. Therefore, the output pulses of AND gates 1, 4 and 5 are in the 0 state, and the output pulses of AND gates 2, 3 and 6 are in the 1 state. FF_1 is reset, FF_2 is set, and FF_3 remains in the 0 state. The 1 stored in FF_1 was shifted into FF_2 and the 0 at the input was shifted into FF_1 . Thus, after two shift pulses, the number stored in the register is 010. The first two bits at the input have been shifted into the register.



WAVEFORMS OF FIGURE 25

Figure 26

During the third shift pulse, the bit applied to the A input is a 1 (and a 0 at B). FF_1 is in the 0 state, FF_2 is in the 1 state, and FF_3 is in the 0 state. Therefore, the output pulses of AND gates 2, 3 and 6 are in the 0 state, but the output pulses of AND gates 1, 4 and 5 are in the 1 state. FF_1 is set, FF_2 is reset and FF_3 is set. The 1 stored in FF_2 was shifted into FF_3 , the 0 stored in FF_1 was shifted into FF_2 , and the 1 at the input was shifted into FF_1 . The number in the register is 101. Thus, after three shift pulses, all three bits at the input have been shifted into the register.

To read-out the register of Figure 25, a 0 is applied to the A input and a 1 is applied to the B input. The clock continues to produce shift pulses. The A input is maintained at a relatively negative dc level (gate 1 has a 0 output pulse), and the B input is maintained at a relatively positive dc level (gate 2 has a 1 output pulse). At the application of the fourth shift pulse, the output pulses of AND gates 2, 3 and 6 are in the 1 state. FF_1 is reset (to the 0 state), FF_2 is set (to the 1 state), and FF_3 is reset. The first bit to the right in the number 101 is available at the register outputs. At the application of the fifth shift pulse, the output pulses of AND gates 2, 4 and 5 are in the

1 state. FF_1 remains in the 0 state. FF_2 is reset, and FF_3 is set. The second bit in the number 101 is available to the register outputs. Finally, at the application of the sixth shift pulse, the output pulses of AND gates 2, 4 and 6 are in the 1 state. FF_1 and FF_2 remain in the 0 state and FF_3 is reset. The last bit at the left in the number 101 is available at the register outputs. After all the flip-flops are read out, they are left in the 0 state. Notice that the bits appeared at the register outputs one after the other, or serially.

The output signals available from FF_3 in Figure 25 are in dc (static) form. The signal at the y_3 output represents the number being read out; the $\bar{y}_3$ output provides an inverted signal of what was read-out. To obtain a pulse output from the register, the y_3 output and the clock (or shift) pulses are applied to the inputs of AND gate 7. AND gate 7 produces an output representing 1 only when the dc level output represents 1 (FF_3 is SET) and a clock pulse occurs. That is, the presence of a pulse at the pulse output represents 1 and the absence of a pulse represents 0.

In Figure 26, the pulses shown with dashes indicate what takes place when two consecutive logic 1's appear at the output of the register shown in Figure 25.

The logic diagram of a left-shift register is the mirror image of Figure 25. (Figure 25 is the logic diagram of a right-shift register.) The inputs of a left-shift register are on the right side of the diagram, the outputs are on the left, and all of the symbols are reversed. Data are shifted to the left instead of to the right; however, the stored number is still read from left to right (in the order of C-B-A).

When high-speed flip-flops are used in a shift register to obtain high-speed operation, a flip-flop may switch from one state to the other before the end of the shift pulse. When this happens, the next flip-flop records the new bit in the first flip-flop instead of the original bit as desired. For example, assume that the register of Figure 25 holds the number 101, and assume that this number is to be shifted one place to the right. When the shift pulse occurs, FF_2 may be set to the 1 state and its outputs may change before the end of the shift pulse. With the shift pulse still applied to AND gate 5, FF_3 would remain in the 1 state when it should be reset to the 0 state. This problem of more than the desired number of flip-flops changing states during a shift pulse is called RACING.

In a low-speed flip-flop, the change in output states does not occur until some time after the input. This is due to the switching time of the circuit. The time interval between the time that the input is applied and the time that the output occurs is a built-in circuit delay. In the circuit of Figure 25, it is

important that the width of the clock pulse be less than the delay caused by one AND gate and one flip-flop in order to prevent "racing." In high-speed flip-flops, the input pulse width can be controlled so that it is less than the built-in circuit delay. However, this usually is expensive or results in critical circuit operation. Another solution to racing is to use flip-flops which change states only on the trailing edges of the input pulses. For example, flip-flops may be used which require positive input pulses, and which change states only at the trailing (or negative-going) portions of the positive input pulses.

Still another way to overcome racing is to delay the flip-flop inputs by placing the contents of each flip-flop in a temporary storage device such as a delay line or another flip-flop. At some time after the shift pulse has ended, the stored information is transferred to the next flip-flop in the register.

In Figure 25, the dashed line boxes show how delay lines may be connected to the flip-flop inputs to overcome racing. When a shift pulse occurs, the contents of the flip-flops are transferred to delay lines. Shortly after the end of the shift pulse, the contents of the delay lines appear at the flip-flop inputs. The delay must be as small as the circuit operation will allow, since this action slows down the circuit operation. Too much delay cancels the speed gained by utilizing high-speed flip-flops.

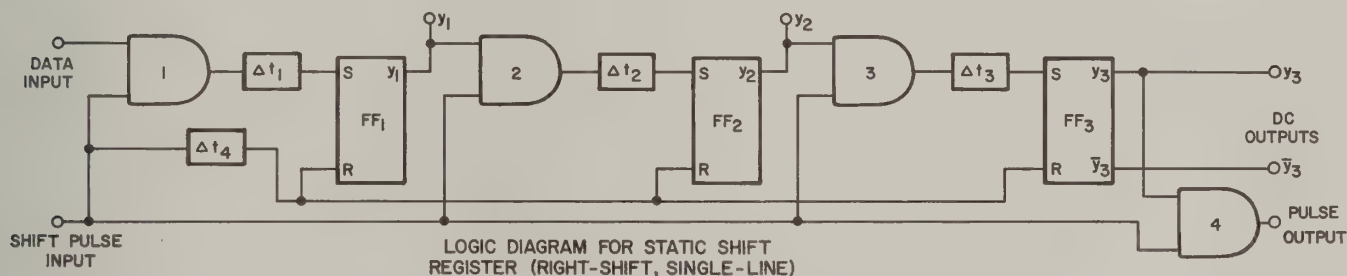
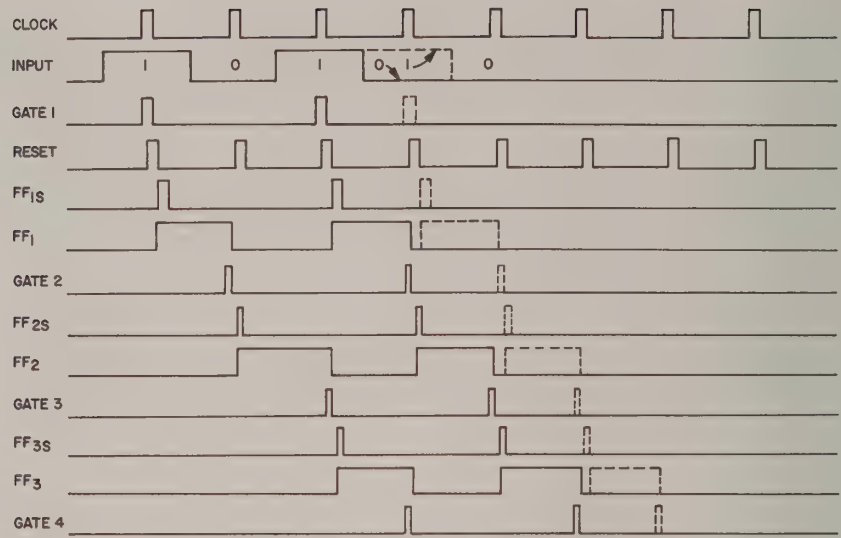


Figure 27

Figure 27 shows the logic block diagram of a register similar to Figure 25; however, only about half as many AND gates and delay lines are used. When a shift pulse occurs in the register of Figure 27, pulses are inserted into delay lines Δt_1 , Δt_2 and Δt_3 , provided the data input or the preceding flip-flops are in the 1 state. A logic 0 pulse is inserted into a delay line if the data input or the preceding flip-flop is in the 0 state. The shift pulse, delayed by Δt_4 , clears all of the flip-flops (to the 0 state), while delay lines Δt_1 , Δt_2 and Δt_3 hold the contents of the register. Figure 28 contains the waveforms of the circuit of Figure 27.

After being cleared, each flip-flop is set (to the 1 state), provided the delay line preceding it contains a logic 1 pulse. If the preceding delay line contains a logic 0 pulse, the flip-flop remains in the 0 state. The delay provided by



WAVEFORMS OF FIGURE 27

Figure 28

Δt_4 must be shorter than the delays provided by Δt_1 , Δt_2 and Δt_3 to insure that the flip-flops are cleared before set pulses from Δt_1 , Δt_2 and Δt_3 appear at their inputs. The output flip-flop and gate of Figure 27 are identical to those of Figure 25.

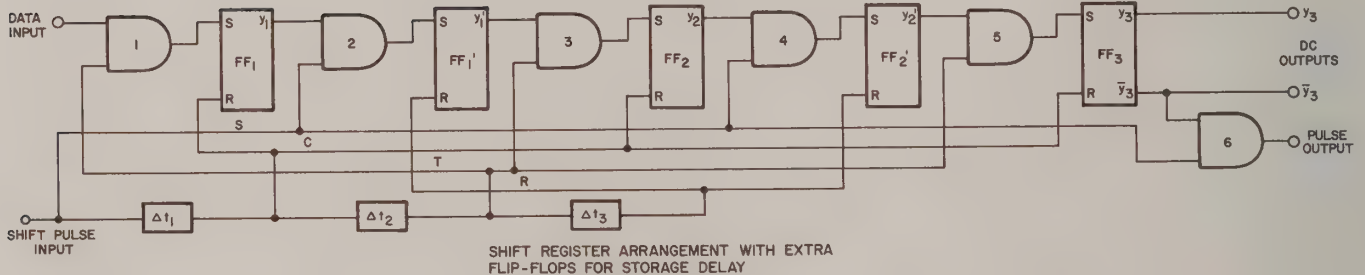


Figure 29

Figure 29 shows a shift register in which two extra flip-flops are used for temporary storage, and Figure 30 shows the waveforms. The three main storage flip-flops are represented by FF, and the two temporary storage flip-flops are represented by FF'. If FF₁ is in the 1 state when a shift pulse (S) occurs, FF₁' is set to the 1 state through AND gate 2. Similarly, if FF₂ is in the 1 state, FF₂' is set to the 1 state through AND gate 4. All main storage flip-flops are then cleared to the zero state by a clear pulse, C. Clear pulse C is simply the clock, or shift, pulse delayed by Δt_1 .

After the main flip-flops are cleared to zero, the contents of each temporary storage flip-flop are transferred to the next main storage flip-flop by a transfer pulse T. Transfer pulse T is a clock pulse delayed by Δt_1 and Δt_2 . The

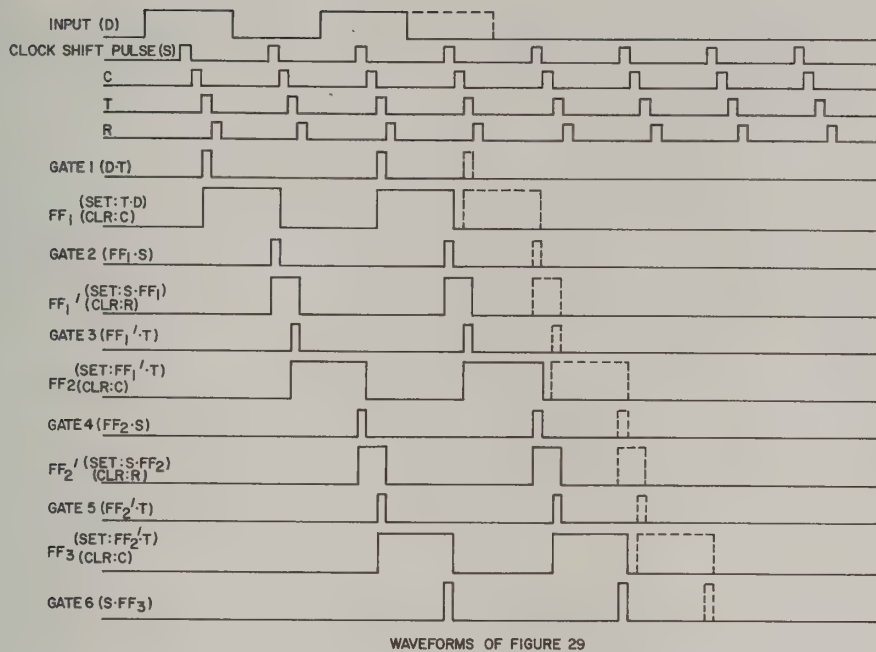


Figure 30

data input, if any, is also transferred to FF₁ by transfer pulse T. A reset pulse R resets the two temporary storage flip-flops after the entire shift operation is complete. Reset pulse R is the clock pulse delayed by Δt_1 , Δt_2 and Δt_3 . The output circuits of this register are identical to those of Figures 25 and 27.

An oscilloscope connected to the y terminal of FF₁, FF₂ or FF₃ in Figure 27 or 29 would reveal gaps between consecutive logic 1 pulses when 1011 is fed into the register. This is shown in Figures 28 and 30.

Dynamic Shift Registers

In a **DYNAMIC SHIFT REGISTER**, information in the form of pulses is applied to the input of a delay line. The output is connected back to the input so that the pulses continually circulate around the loop. This permits retention of information that would be lost at the output of a serial data transmission system that does not utilize feedback. Figure 31 shows a shift register which uses dynamic storage techniques. Delay lines Δt_1 and Δt_2 together provide a one word-time delay; therefore, the register has a capacity of one word. The recirculating loop is made up of the shift circuits, AND gate 2, and OR gate 1. An ordinary flip-flop shift register could be used in place of Δt_1 , if the bits are continually shifted through the flip-flops to provide the same effect as the delay line.

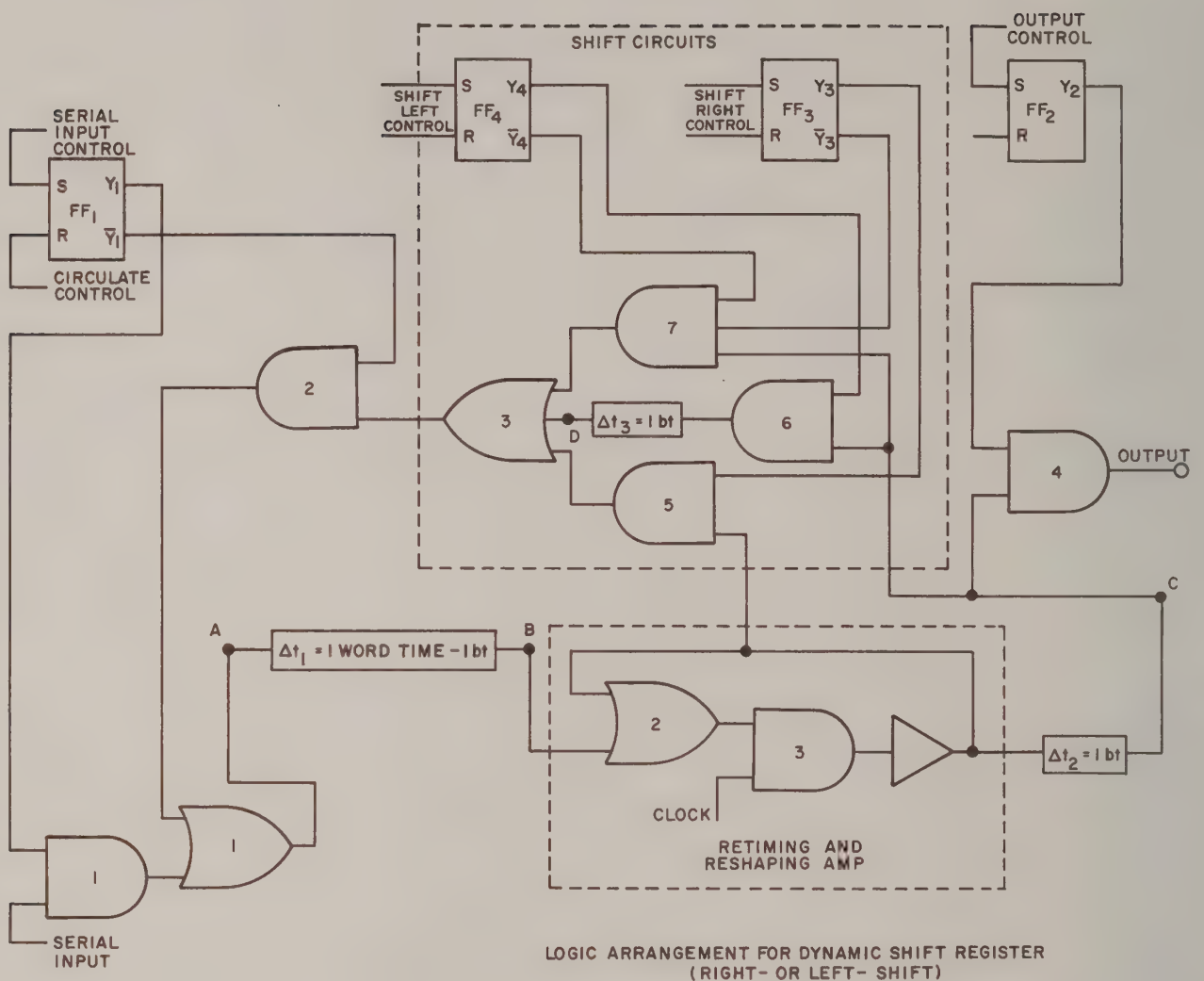


Figure 31

In Figure 31, flip-flops FF₁ through FF₄ are actually part of the control system rather than part of the register. If FF₁ is set to the 1 state, pulses representing information in serial form can pass through AND gate 1 and OR gate 1 to the input of delay line Δt_1 . From the output of Δt_1 , the pulses pass through a retiming and reshaping amplifier and a one bit-time delay Δt_2 . Since FF₁ is in the 1 state, no pulses can pass through AND gate 2.

Once the desired information is inserted into the register through the serial input, FF_1 is reset. Now AND gate 1 is disabled and AND gate 2 enabled. The pulses at the output of Δt_2 pass through the shift circuits, AND gate 2, and OR gate 1 back to the input of Δt_1 . The pulses continually circulate around the loop until the register is cleared by setting FF_1 , thus disabling

AND gate 2. To read the information stored in the register, FF_2 is set. This enables AND gate 4, allowing the pulses to pass through to the output of the register.

It is necessary to include a retiming and reshaping amplifier in the recirculating loop since the delay time of the register may not exactly correspond to the timing of the system. If a number of dynamic shift registers are used in a digital system, the pulses passing through the various registers will drift slowly out of synchronization, unless retiming and reshaping amplifiers are used. The delay lines in a dynamic register also attenuate and distort the input pulses. The retiming and reshaping amplifier restores these pulses to their original amplitude and shape. Since Δt_2 is placed after the retiming and reshaping amplifier in the register of Figure 31, it must be carefully chosen to provide an accurate delay of one bit-time and produce as little attenuation and distortion as possible.

In the register of Figure 31, pulses at the output of Δt_1 pass through OR gate 2 to one input of AND gate 3. These pulses are timed to arrive at AND gate 3 slightly before the corresponding clock pulses. When a pulse from Δt_1 and also a clock pulse appear at the AND gate 3 inputs, an output pulse from AND gate 3 passes through the amplifier, and appears at the input to Δt_2 . The output of the amplifier is also fed back to one input of OR gate 2. Therefore, even though the output pulse from Δt_1 terminates before the trailing edge of the clock pulse occurs, the feedback path through OR gate 2 to AND gate 3 maintains the input to the amplifier as long as the clock pulse is present. When the clock pulse terminates, no signal is applied to the lower input of AND gate 3, and the input to Δt_2 also terminates. This permits the entire clock pulse to appear at the input to Δt_2 .

Because of the circulating data paths in dynamic registers, data shifting in the dynamic register is accomplished somewhat differently than in the static register. In the dynamic register, a word is stored in a dynamic device and the shift is accomplished by delaying or advancing the word in relation to the other words in the system. Figure 32 shows how this is done for a five-bit word. Three delay lines, Δt_1 , Δt_2 and Δt_3 , are connected in series. Delay line Δt_1 has a delay time of one word-time minus one bit-time (4 bt), and Δt_2 and Δt_3 both have delay times of one bit-time. Thus, Δt_1 and Δt_2 together provide a delay time of one word-time (5 bt). All three delay lines provide a total delay time of one word-time plus one bit-time (6 bt). Note: 5 bt means that there are 5 bit-times associated with the delay times.

Figure 32 shows the pulse waveforms and the corresponding binary numbers for three word-times at four points along the signal path. The presence of a pulse represents 1 and the absence of a pulse represents 0. During the first word-time, pulses representing the five-bit binary number 00111 are applied to Point A. During the second word-time, this same number appears at

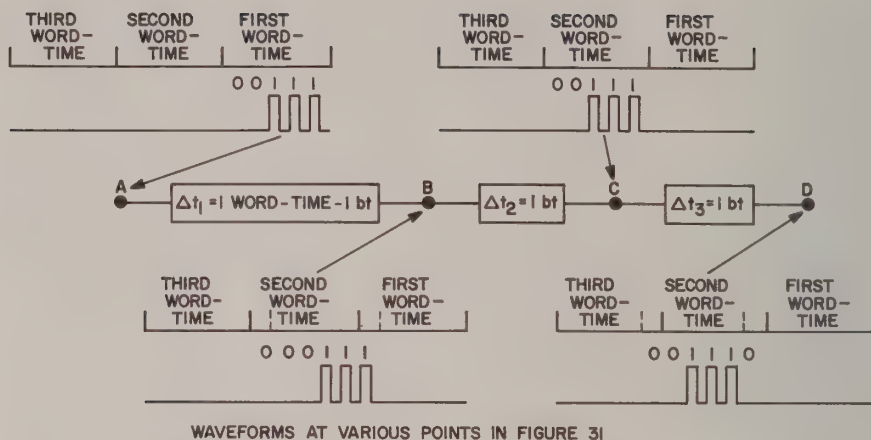


Figure 32

Point C since Δt_1 and Δt_2 provide a delay of exactly one word-time. The delay from Point A to Point B is one bit-time less than a complete word-time, or 4 bt. Therefore, during the fifth bit-time of the first word-time, the first pulse in the word appears at Point B, followed by the other bits in the word at successive bit intervals. This word then passes into Δt_2 . At the beginning of the second word-time, the first pulse has already passed from Δt_1 to Δt_2 and is beginning to appear at Point C. The second pulse is beginning to appear at Point B. This is the same as shifting the number one place to the right.

The delay from Point A to Point D is one bit-time greater than one word-time. At the beginning of the second word-time, the first pulse is still at Point C and just entering Δt_3 . Therefore, the first pulse does not appear at Point D until one bit-time after the second word-time begins. This is the same as shifting the number one place to the left. Thus, by taking the output at Point B or Point D, a right or left shift, respectively, can be obtained.

The shift circuits in the register of Figure 31 cause the pulses to bypass Δt_2 when a right shift is desired, and Δt_3 is inserted in the signal path when a left shift is desired. To see how this is accomplished, first trace out the signal path when no shift is desired. Flip-flops FF_1 , FF_3 and FF_4 are in the 0 state. The pulses enter Δt_1 at the left, pass through the delay line, through the retiming and reshaping amplifier, and through Δt_2 . They arrive at the output of Δt_2 exactly one word-time after entering Δt_1 . Since both FF_3 and FF_4 are in the 0 state, the pulses pass through AND gate 7. From here they pass through OR gate 3, AND gate 2 (FF_1 is also in the 0 state), and OR gate 1 back to the input of Δt_1 .

To shift the number to the right, FF_3 is set automatically by the control system. This disables AND gate 7 and enables AND gate 5, and Δt_2 no

longer is part of the recirculating loop. The loop now contains Δt_1 , the retiming and reshaping amplifier, AND gate 5, OR gate 3, AND gate 2, and OR gate 1. Since Δt_2 is no longer in the loop, the total delay time is equal to the delay of Δt_1 , or one word-time minus one bit-time. This shifts the number one place to the right. If FF_3 of Figure 31 remains in the 1 state for three word-times, then the word is shifted three places to the right, once for each time around the loop. To end the right-shift operation, FF_3 is reset to the 0 state.

To shift the number to the left, FF_4 is set. This disables AND gate 7 and enables AND gate 6, placing Δt_2 and Δt_3 in the recirculating loop. The loop now contains Δt_1 , the retiming and reshaping amplifier, Δt_2 , AND gate 6, Δt_3 , OR gate 3, AND gate 2, and OR gate 1. Since Δt_3 is now in the loop, the total delay is equal to the sum of the delays of Δt_1 , Δt_2 and Δt_3 , or one word-time plus one bit-time. This shifts the number one place to the left. The number shifts one place to the left each time around the loop as long as FF_4 is in the 1 state. To end the left-shift operation, FF_4 is reset.

Parallel Addition and Subtraction

Addition and subtraction can be accomplished at a much faster rate using **PARALLEL ADDITION** or **PARALLEL SUBTRACTION**, as opposed to the serial addition or serial subtraction you are already familiar with. In the parallel operations, all inputs are handled simultaneously and outputs generated simultaneously, rather than one bit position at a time. The "expense" of the increased speed is the use of a more complex logic arrangement, requiring many more components.

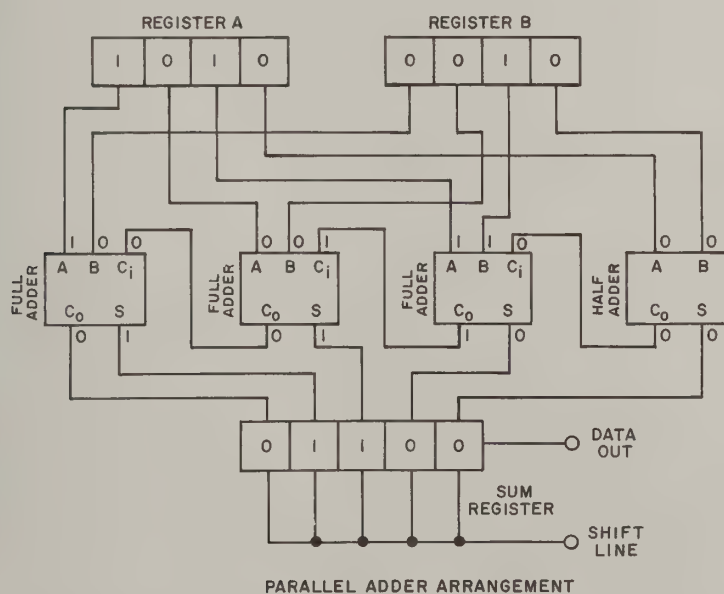


Figure 33

REGISTER AND COUNTER CIRCUITS

Inputs		Outputs	
A	B	S	C _O
1	1	0	1
1	0	1	0
0	1	1	0
0	0	0	0

Half-Adder Truth Table

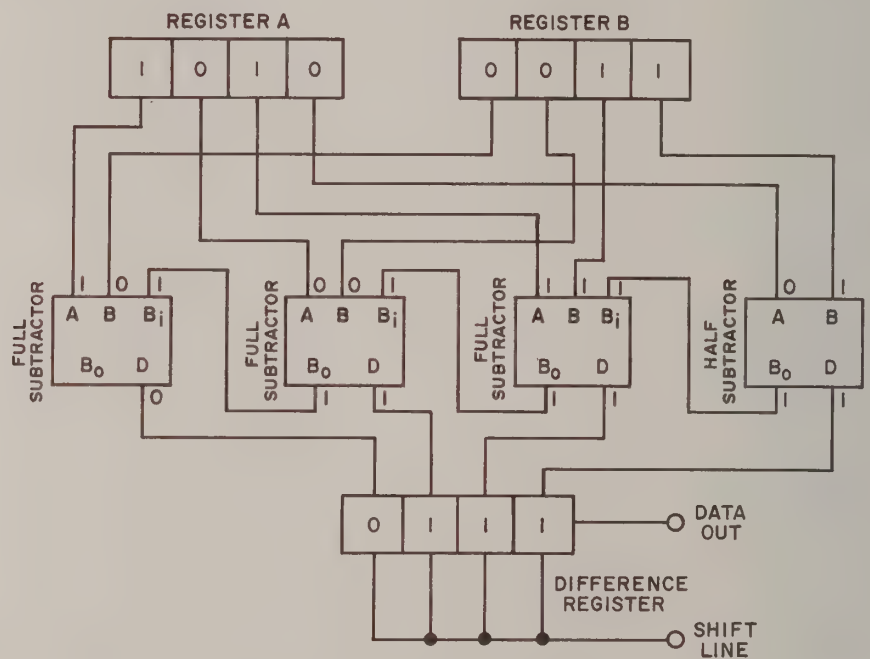
Table 3

Inputs			Outputs	
A	B	C _i	S	C _O
1	1	1	1	1
1	1	0	0	1
1	0	1	0	1
1	0	0	1	0
0	1	0	1	0
0	0	1	1	0
0	1	1	0	1
0	0	0	0	0

Full-Adder Truth Table

Table 4

The logic for parallel addition is shown in Figure 33. First, the two numbers to be added are stored in registers A and B. Register A contains decimal 10 (binary 1010), and register B contains decimal 2 (binary 0010). The bits are then transferred to the adders. The output bits are stored in the sum register (which contains decimal 12), where they can be shifted to some other unit. The sum register is one-bit-position larger than the binary numbers being added. This allows the sum to "hold" the higher order carry which may be generated by (in this case) the third full adder. The sample addition shown in Figure 33 can be traced through the arrangement with the aid of the truth data in Tables 3 and 4.



PARALLEL SUBTRACTOR ARRANGEMENT

Figure 34

Figure 34 is a parallel subtractor arrangement. The general arrangement is similar to that of Figure 33, except that subtractors have replaced the adders and the output register has the same number of bits as the input registers. The sample operation can be traced with the aid of the truth data in Tables 5 and 6.

Inputs		Outputs	
A	B	D	B _O
1	1	0	0
1	0	1	0
0	1	1	1
0	0	0	0

Half-Subtractor Truth
Table (A - B)

Table 5

Inputs			Outputs	
A	B	B _i	D	B _O
1	1	1	1	1
1	1	0	0	0
1	0	1	0	0
1	0	0	1	0
0	1	0	1	1
0	0	1	1	1
0	1	1	0	1
0	0	0	0	0

Full-Subtractor Truth Table
(A - B)

Table 6

Parallel-Serial/Serial-Parallel Conversions

Different portions of a digital system may use either serial or parallel data formats; therefore, it becomes necessary, at the interface between these portions, to convert from one form to another. Shift registers are commonly used to accomplish this. For example, the output registers in Figures 33 and 34 are shown as parallel-to-serial conversion registers. The sum digits and difference digits are generated in the parallel format, but are shifted out in serial format.

Figure 35 illustrates the logic circuit arrangement for an available IC RTL serial-parallel shift register. This logic utilizes four type D flip-flops. The PRECLEAR input is a reset signal to clear the register of any stored information. When the PRECLEAR and PARALLEL ENABLE signals are absent (logical zero), the negative transitions of the CLOCK signal cause a logical 1 output from the NOR gate which goes to the gating terminals, G, of the flip-flops, to allow transfer of SERIAL INPUTS data from pin 7. This data is passed from flip-flop to flip-flop on subsequent negative transitions of the CLOCK signal. Thus, it takes four clock pulses to enter a four-bit binary code into the register. Another four clock pulses can shift

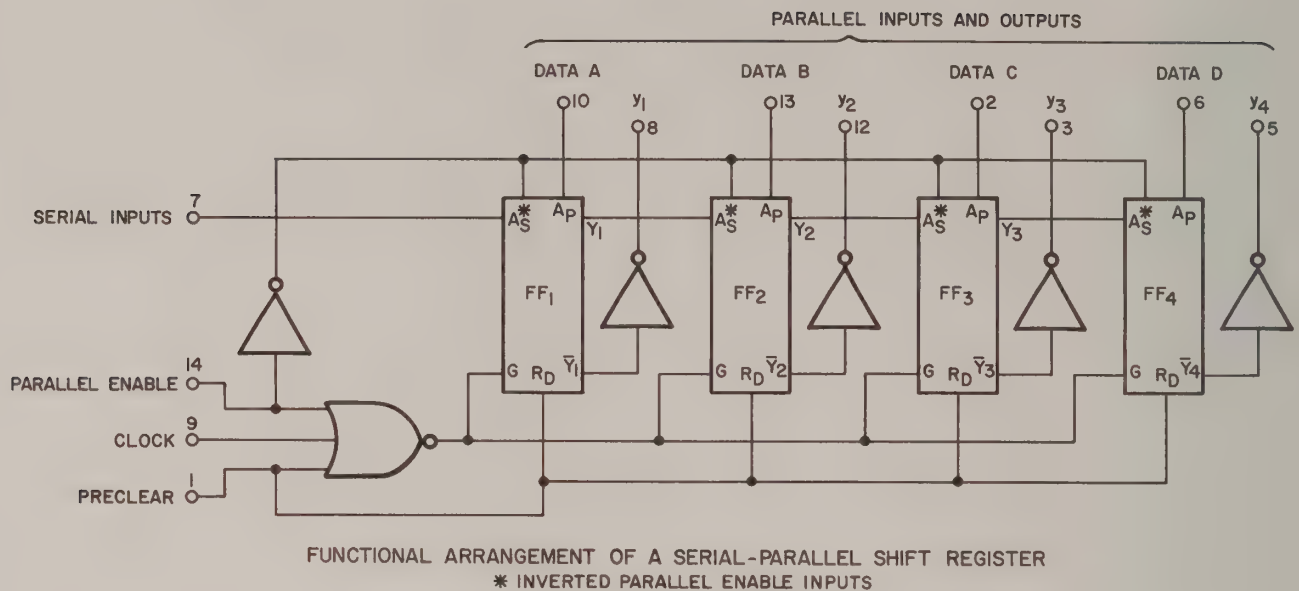


Figure 35

the four-bit code out of the register (at pin 5), even if the data has been entered in the parallel mode. Parallel mode data can enter the flip-flop A_P terminals only when the PARALLEL ENABLE signal is in a logical 1 state. The PARALLEL ENABLE signal is applied (through an inverter) to gating terminals of the flip-flops that are indicated by asterisks. This condition also inhibits the clock pulse output of the NOR gate. This register differs from the arrangements described previously in another respect: successive flip-flops are fed by the y outputs, rather than the $\bar{y}$ outputs. In turn, the flip-flop states are read out at the $\bar{y}$ terminals through inverter stages. Actually, the y states are read out at pins 3, 5, 8 and 12, as required. Because of these features, this IC is an extremely versatile unit. This particular IC circuit can be utilized as an accumulator or a buffer register, as well as a serial-parallel/parallel-serial converter.

Multiplication Logic

One approach to performing multiplication in a digital computer is to use the process of repeated addition. One of the factors is added to itself until the number of times is equal to the other factor: $0101 \times 0100 = 0101 + 0101 + 0101 + 0101 = 10100$. The first factor (0101) is added to itself four (0100) times. In the arithmetic unit, the number of additions can be counted until the second factor is reached. A down-counter is frequently used. It is preset to the second factor, and then counts down each time a clock pulse occurs.

The following Practice Exercise questions cover the subjects which you have just studied. They are:

SHIFT REGISTERS

STATIC SHIFT REGISTERS

DYNAMIC SHIFT REGISTERS

PARALLEL ADDITION AND SUBTRACTION

24. Which type of shift (a right or left) doubles a binary number?
25. If a binary number, stored in a shift register, is to be divided by 16 (decimal), the binary number must be shifted (a) _____ places to the (b) _____.
26. Without performing the long-hand multiplication, write the binary product of 1001 and 100.
27. If FF_2 of Figure 25 were in the 1 state, which AND gate, AND gate 5 or 6, would be in the 1 state during the next shift pulse?
28. What type of output (serial or parallel) is obtained when shift pulses are applied to the shift register of Figure 25?
29. At the pulse output of Figure 25, the presence of a pulse represents 1 and the absence of a pulse represents 0. True or False?
30. Delay lines are required in the shift register of Figure 25 when low-speed flip-flops are used. True or False?
31. In the shift register of Figure 25, the delay lines store the flip-flop outputs until (a) just before the end of the shift pulse, (b) after the shift pulse has ended, (c) the next shift pulse occurs.
32. When the output pulse of Δt_1 in Figure 31 terminates, the output pulse of the retiming and reshaping amplifier also terminates. True or False?
33. The width of a pulse appearing at the input to Δt_2 (Figure 31) is the same as the width of (a) the output pulse of Δt_2 , (b) the clock pulse.
34. Delaying a word for one bit-time less than a complete word-time (Figure 31) provides what type of shift (right or left)?

35. When FF_3 and FF_4 of Figure 31 are in the 0 state, the pulses pass through AND gate 7 and no shift occurs. True or False?
36. When FF_4 of Figure 31 is in the 1 state, Δ_3 is inserted in the recirculating loop and the number is shifted to the left. True or False?

The clock pulses are also used to sequence the repeated additions. When the down-counter reaches all -0's, the process is stopped and the result is read out of the sum register.

If you were to perform pencil and paper multiplications in this manner, you would soon give up—especially when large values are involved. However, for a high-speed digital computer, it is a simple and fast operation.

The most frequent method used to implement binary multiplication utilizes the shifting technique. This requires more complex circuitry; however, it is faster than repeated addition:

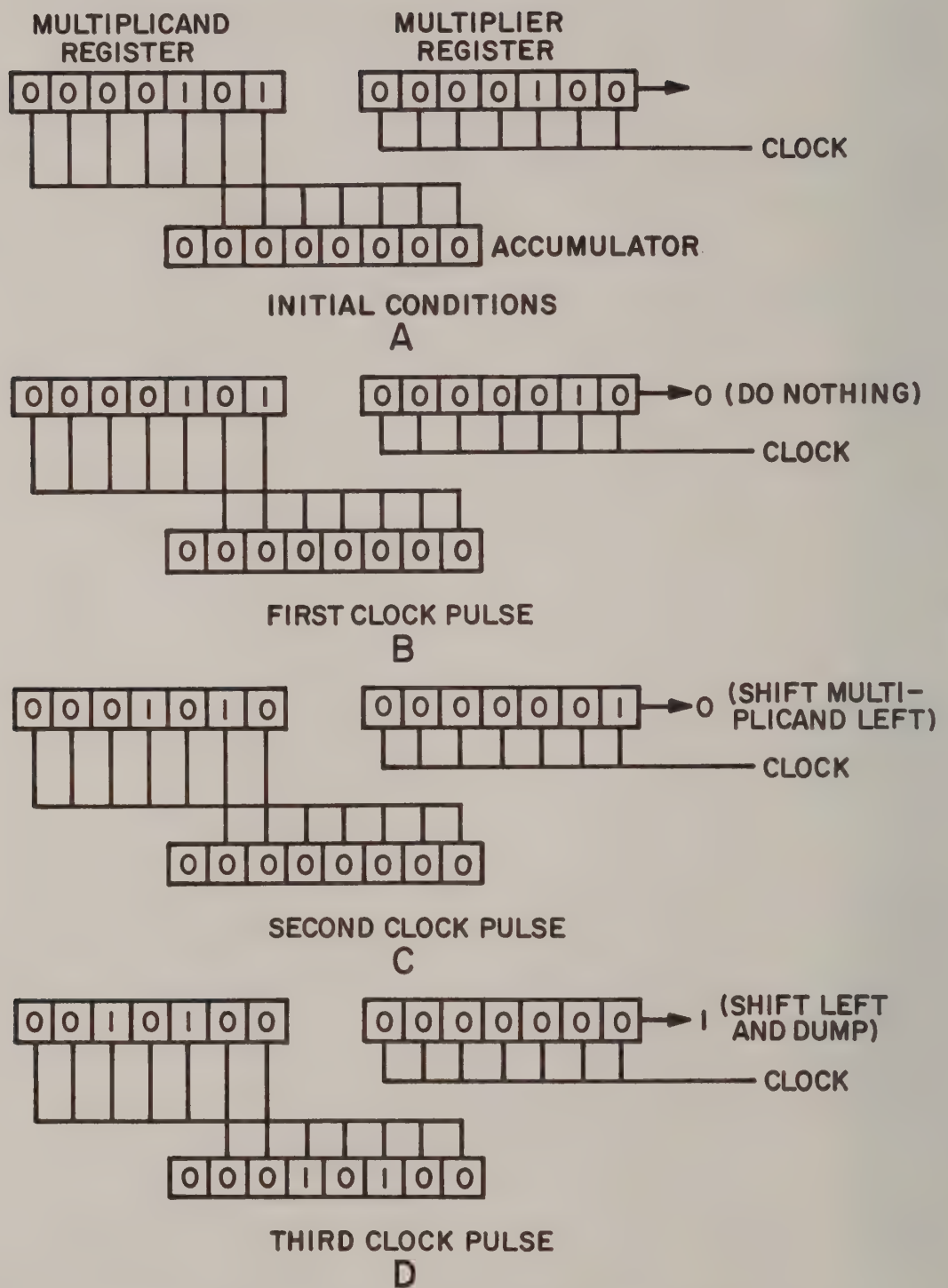
$$\begin{array}{r}
 0101 \\
 \times 0100 \\
 \hline
 0000 \\
 0000 \\
 0101 \\
 0000 \\
 \hline
 0010100
 \end{array}$$

Notice that each successive partial product is shifted one additional digit position to the left. The value of each partial product is 0 if the multiplier bit is 0, and a repeat of the first factor if the multiplier bit is 1. (Except for the two-place shift brought on by the first two, the first, second, and fourth partial product could have been ignored.)

Figure 36A illustrates how the procedure can be implemented with shift registers. Various logic arrangements can be used. Each clock pulse shifts the multiplier to the right, outputting the next higher order binary digit. If the multiplier output digit is a 0, the multiplicand data is shifted one bit to the left (except on the first clock pulse). If this digit is a 1, in addition to the shift, the multiplicand data is added to the value stored in the accumulator register. The process is stopped after the number of clock pulses equals the number of bits in the multiplier, and the product is transferred out of the accumulator (not shown in Figure 36).

Division Logic

Division can be performed by repeated subtraction. Repeated subtraction is performed until there is no remainder, or until a specific number of subtractions are performed, depending upon the degree of accuracy required. An up-counter can keep track of the subtractions. Its output can be accumulated to represent the binary quotient.



MULTIPLIER OPERATION WITH SHIFTING TECHNIQUE

Figure 36

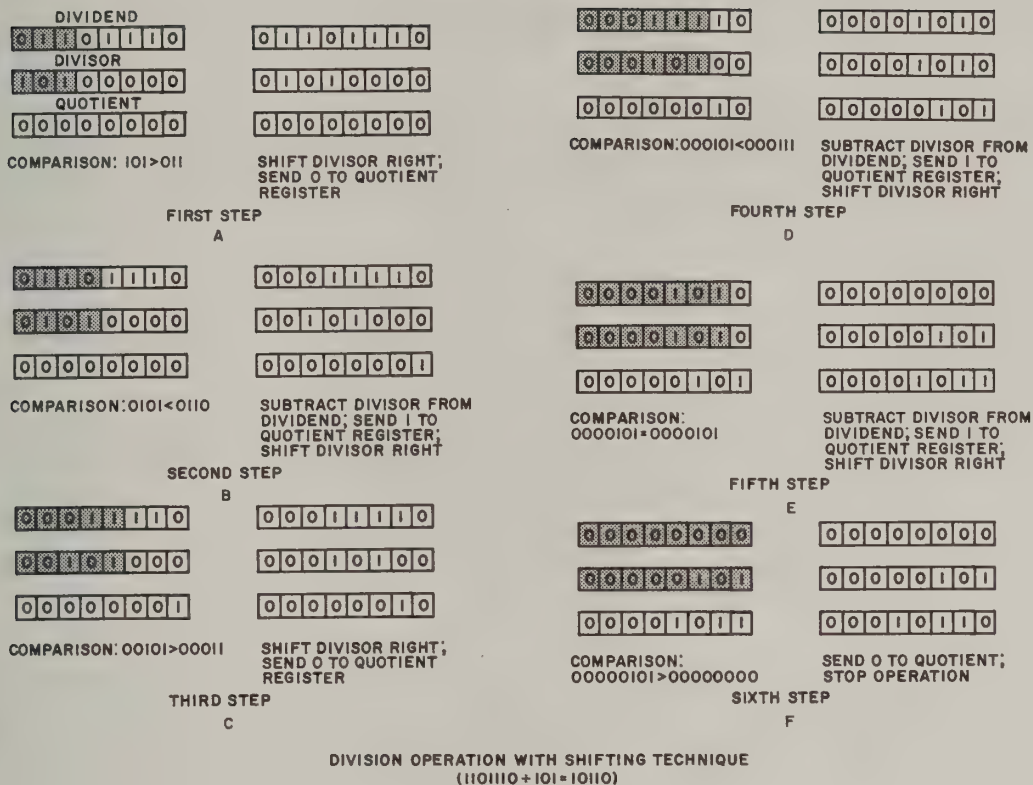


Figure 37

A more common division technique also utilizes the shifting procedure. To accomplish this action, the dividend is entered into a register and the significant divisor bits are entered in higher order positions in another register. This is shown in Figure 37A. Now, the higher order bits are compared. If the divisor bits are less than or equal to the dividend bits, a subtraction is performed, and a 1 is sent to the right-most position in the quotient register, and the divisor bits are shifted one position to the right. If the divisor bits are greater than the dividend bits (as in Figure 37A), the divisor bits are shifted one position right and a 0 is sent to the quotient register. Whenever a 1 or a 0 is sent to the quotient register, the 1's and 0's in that register are shifted one space to the left to provide a space for the new 1 or 0. The process is continued until the significant divisor bits reach the lowest order positions.

Sometimes division does not come out "even;" therefore, fractional registers must also be included. The divisor shift would then be continued until the desired accuracy is attained.

46. Show the first step in the division of Practice Exercise 45 in the registers below.

DIVIDEND							
DIVISOR							
QUOTIENT							

DIVIDEND							
DIVISOR							
QUOTIENT							

47. Show the second step in the division of Practice Exercise 45 in the registers below.

DIVIDEND							
DIVISOR							
QUOTIENT							

DIVIDEND							
DIVISOR							
QUOTIENT							

48. Show the third step in the division of Practice Exercise 45 in the registers below.

DIVIDEND							
DIVISOR							
QUOTIENT							

DIVIDEND							
DIVISOR							
QUOTIENT							

49. Show the fourth step in the division of Practice Exercise 45 in the registers below.

DIVIDEND							
DIVISOR							
QUOTIENT							

DIVIDEND							
DIVISOR							
QUOTIENT							

50. Show the fifth step in the division of Practice Exercise 45 in the registers below.

DIVIDEND							
DIVISOR							
QUOTIENT							

DIVIDEND							
DIVISOR							
QUOTIENT							

stage flip-flops in cascade, since there are seven digits. This form of counting is directly compatible with the BCD code.

When cascaded flip-flops are used for the temporary storage of data, they are known as registers. Static registers work with longer duration dc signal levels. Pulse or dynamic registers work with short duration pulses.

In a static shift register, a number may be shifted to either the right or the left. A shift to the right is the same as halving the stored binary number or dividing by 2. A shift to the left is the same as doubling the stored binary number or multiplying by 2. A number may be inserted into a static shift register in parallel form and shifted out in serial form to provide a parallel-to-serial conversion. Or the number may be inserted into a static shift register in serial form and shifted out in parallel form to provide a serial-to-parallel conversion.

In a dynamic register, a series of pulses represents the stored number. The bits (or pulses) in the number continually change position in space as in a delay line. The magnitude of a bit is indicated by the time position of the pulse which represents it. When a number is read out of a dynamic storage device, the first bit at the right in the number (the least significant bit) may appear during the first bit-time; the second bit during the second bit-time; the third bit during the third bit-time; etc. This is called storage in time. To obtain a right shift in a dynamic shift register, the word must be delayed one bit-time less than a complete word-time. To obtain a left shift in a dynamic shift register, the word must be delayed one bit-time more than a complete word-time.

Registers are used extensively in digital computer applications. Data may be clocked into a shift register at one rate, and clocked out at another rate. Here, the register acts as a buffer between different parts of the system operating at different speeds. Registers can also be used as serial-to-parallel and parallel-to-serial converters.

In arithmetic units, registers are used as adders, subtractors, multipliers and dividers. Combinations of registers, half adders and full adders (or half subtractors and full subtractors) can be used to form parallel adders (or subtractors). Since all bits are handled in a simultaneous rather than sequential manner, the operations can be performed very rapidly.

Binary multiplication can be performed by a process of repeated addition or by using a shift technique. Binary division can be performed by repeated subtractions or by using a shift technique. Of the two techniques, shifting is the more complex, but it is also faster.

IMPORTANT DEFINITIONS

ACCUMULATOR—A unit which is able to store a number, add another number to the stored value, and store the sum.

BINARY COUNTER—A group of flip-flops which counts the number of input pulses or state changes. Each flip-flop in the group represents a binary number position.

BIT TIME—One clock cycle in a synchronized digital system.

BUFFER—A storage device where information is fed in at one rate and read out at another rate. Buffers are usually used to match the speed of various units in a digital system. Not to be confused with buffer stages, buffer amplifiers or buffer gates. To avoid the confusion, these storage devices are sometimes called **BUFFER REGISTERS**.

CLOCK—The pulse generator which generates the basic timing signals in a digital system.

COUNTERS—Electronic or mechanical devices used to count electrical or mechanical events.

DATA WORD—A group of bits which represents a basic unit of information processed in a digital system. The actual length varies from system to system.

DECADE COUNTER—A group of flip-flops and gates which accepts a single serial input line and counts the number of times the input changes state, outputting a group of bits representing the decimal digits from 0 through 9. The decade counter may also have a parallel input lead for each flip-flop and a serial/parallel input mode control lead.

DECADE REGISTER—A group of two or more **DECADE COUNTERS** connected in series, outputting a group of bits representing the decimal digits from 0 through 99, 0 through 999, or 0 through 9999, etc.

DECISION TIME—An interval during which logical operations are performed in a synchronized digital system. The **DECISION TIME** is usually equal to one-half **BIT-TIME** (when the clock cycle is in its 1 state).

DIGIT TIME—The interval associated with a specific group of bits within a **DATA WORD**. The **DIGIT TIME** varies from system to system.

DOWN-COUNTER—A counter in which the count decreases by a fixed amount with each input pulse. The **DOWN-COUNTER** is also known as a backward counter.

IMPORTANT DEFINITIONS (Continued)

DYNAMIC LOGIC SYSTEMS—Logic systems in which ac signals or pulses are used to represent bits of information.

DYNAMIC SHIFT REGISTER—A SHIFT REGISTER where the bits continually change position. See STATIC SHIFT REGISTER for contrast.

PARALLEL ADDITION—A process where each bit position, in two binary numbers, is added together in a simultaneous, rather than sequential, manner.

PARALLEL SUBTRACTION—A process where each bit position, in one binary number, is subtracted from the corresponding bit position in another binary number simultaneously, rather than sequentially.

PARALLEL TRANSMISSION—A transfer or handling of the bits in a DATA WORD on a simultaneous, rather than sequential, basis.

PROGRAM—A set of instructions which guides a digital system through its logical procedures.

PULSE LOGIC SYSTEMS—See DYNAMIC LOGIC SYSTEMS.

RACING—A problem created when a high-speed flip-flop changes states during an input pulse, resulting in an erroneous flip-flop state change.

REGISTER—A device which stores information in a digital system. Registers usually have a capacity of one word.

RETIMING AND RESHAPING AMPLIFIER—An amplifier which develops an output signal whose width is determined by the clock pulse and shape determined by the amplifier.

SERIAL TRANSMISSION—A transfer or handling of the bits in a DATA WORD on a one-bit-at-a-time sequential basis.

SHIFT REGISTER—A register which has provisions for shifting the stored number to the right or the left with respect to the decimal point.

STATIC LOGIC SYSTEMS—Logic systems which use dc signals to represent bits of information.

STATIC SHIFT REGISTER—A SHIFT REGISTER where the bits do not change position. See DYNAMIC SHIFT REGISTER for contrast.

UP-COUNTER—A counter in which the count increases by a fixed amount

IMPORTANT DEFINITIONS (Continued)

with each input pulse. The UP-COUNTER is also known as a forward counter.

WORD TIME—The interval associated with a DATA WORD. The WORD TIME varies from system to system. The time required to transport one word from one particular storage device to another.

PRACTICE EXERCISE SOLUTIONS

1. In the data word, each bit is accurately synchronized with the clock pulses.
2. (a) digit pulses.
3. (b) word pulses.
4. False—AND gates synchronize the clock and input signals before they are applied to the OR gate.
5. (a) 10 nanoseconds—One-half the bit time. (b) 160 nanoseconds—8 bits $\times$ 20 nanoseconds. (c) 640 nanoseconds—4 digit times. (d) 32—8 bits per digit $\times$ 4 digits per word.
6. False—Different parts of a system operate at different speeds, implying different basic bit times.
7. True
8. True
9. retiming and reshaping amplifiers.
10. (b) width of the clock pulse.
11. Once the amplifier output has “started” through the simultaneous application of an input pulse and a clock pulse, the feedback pulse will assure that the output duration equals the clock duration, even if the input pulse is removed.
12. A fifth pulse causes y_1 to switch to 1. The 0 output at $\bar{y}_1$ has no effect on the second stage; therefore, there is no change in state for the third or fourth stages. This provides a count of 0101 (decimal 5).
13. The number of flip-flop stages establishes the maximum count for a binary up-counter.
14. For the up-counter, the cascade is formed by feeding the $\bar{y}$ output of one stage to the input of the next stage. For the down-counter, the y output is used for stage-to-stage signal transfer.
15. The counter is cleared by applying a signal to the PRECLEAR (R_D) terminal, making the count 0000. A 1 is then applied to the first-stage (right-most) S_{D1} terminal and to the third-stage S_{D3} terminal, providing a count of 0101 (decimal 5).
16. At the count of decimal 9, the binary count stands at 1001 for the four-stage outputs. The next count causes an output of 1 from the AND gate. At the start of the tenth input pulse, FF_2 goes set for only a very short time

PRACTICE EXERCISE SOLUTIONS (Continued)

because the AND gate has an output pulse which resets FF_2 and FF_3 . Then, $\bar{y}_4$ goes to a logical 1, triggering the succeeding decade counter into the 0001 condition. (See Figure 19.)

17. Seven—One decade counter for each digit in the largest number to be counted.
18. 268,435,456 (or 2^{28})—Each binary counter has four flip-flop stages; therefore, there are 28 flip-flop stages available (7 counters $\times$ 4 flip-flops per counter). Since each stage doubles the count of the preceding stage, and the count starts at one, the result is as shown.
19. False—All register flip-flops are reset (0 state).
20. False—No clear (reset) pulse is required.
21. 01001100
22. True
23. (a) one word; (b) one bit
24. A left shift.
25. (a) four; (b) right
26. 100100—100 is decimal 4; 1001 is multiplied by shifting the number left by two places.
27. AND gate 5.
28. Serial.
29. True
30. False—Delay lines are required to overcome race problems when high-speed flip-flops are used.
31. (b) after the shift pulse has ended.
32. False—The output pulse of the retiming and reshaping amplifier does not terminate until the clock pulse does.
33. (b) the clock pulse.

PRACTICE EXERCISE SOLUTIONS (Continued)

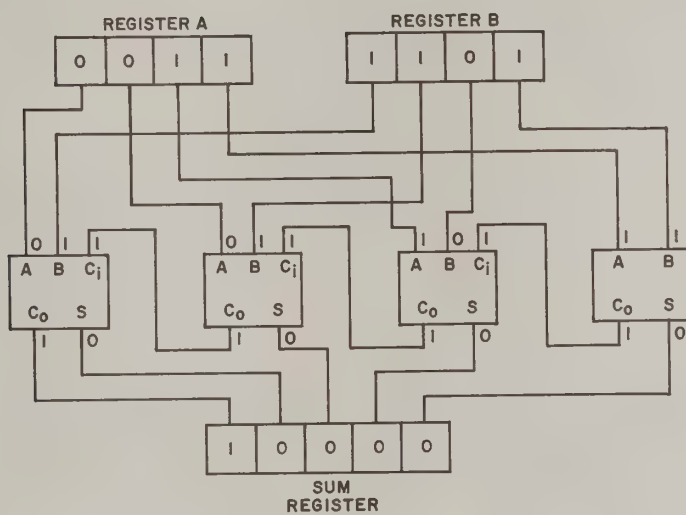
34. A right shift.

35. True

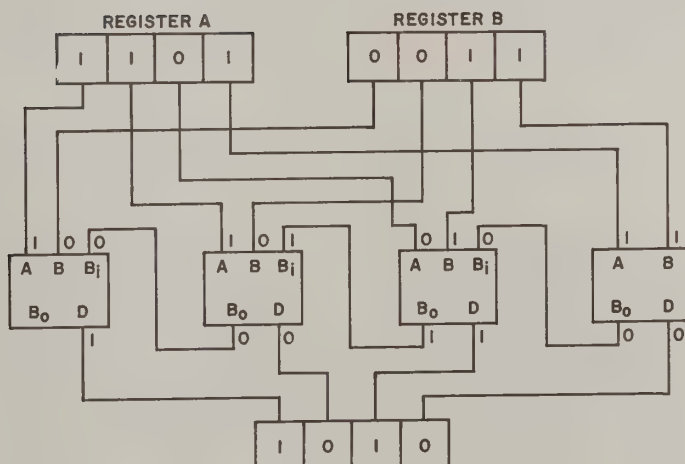
36. True

37. True

38.



39.



40. True

41. Nothing—The data is not shifted on the first clock pulse and, since the first multiplier bit is 0, the data is not added in the accumulator.

PRACTICE EXERCISE SOLUTIONS (Continued)

42. The data is shifted one place to the left to read 11100, and is placed in the accumulator.
43. The data is shifted one more place to the left to read 111000, and is added to the accumulator value.
44. 1010100—The accumulator addition is $11100 + 111000$.

45.

DIVIDEND							
0	1	0	1	0	1	0	0
DIVISOR							
1	1	1	0	0	0	0	0
QUOTIENT							
0	0	0	0	0	0	0	0

46.

DIVIDEND							
0	1	0	1	0	1	0	0
DIVISOR							
1	1	1	0	0	0	0	0
QUOTIENT							
0	0	0	0	0	0	0	0

COMPARISON:
1110 > 0101

DIVIDEND							
0	1	0	1	0	1	0	0
DIVISOR							
0	1	1	1	0	0	0	0
QUOTIENT							
0	0	0	0	0	0	0	0

SHIFT DIVISOR RIGHT;
SEND 0 TO QUOTIENT
REGISTER

47.

DIVIDEND							
0	1	0	1	0	1	0	0
DIVISOR							
0	1	1	1	0	0	0	0
QUOTIENT							
0	0	0	0	0	0	0	0

COMPARISON:
01110 > 01010

DIVIDEND							
0	1	0	1	0	1	0	0
DIVISOR							
0	0	1	1	1	0	0	0
QUOTIENT							
0	0	0	0	0	0	0	0

SHIFT DIVISOR RIGHT;
SEND 0 TO QUOTIENT
REGISTER

PRACTICE EXERCISE SOLUTIONS (Continued)

48.

DIVIDEND

1	0	1	0	1	1	0	0
---	---	---	---	---	---	---	---

DIVISOR

1	0	1	1	0	0	0	0
---	---	---	---	---	---	---	---

QUOTIENT

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

COMPARISON:
 001110 < 010101

DIVIDEND

0	0	0	1	1	1	0	0
---	---	---	---	---	---	---	---

DIVISOR

0	0	0	1	1	1	0	0
---	---	---	---	---	---	---	---

QUOTIENT

0	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---

**SUBTRACT DIVISOR FROM
 DIVIDEND ; SEND 1 TO
 QUOTIENT REGISTER ;
 SHIFT DIVISOR RIGHT**

49.

DIVIDEND

0	0	0	1	1	0	0	0
---	---	---	---	---	---	---	---

DIVISOR

0	0	0	1	1	0	0	0
---	---	---	---	---	---	---	---

QUOTIENT

0	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---

COMPARISON:
 000111 = 000111

DIVIDEND

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

DIVISOR

0	0	0	0	1	1	1	0
---	---	---	---	---	---	---	---

QUOTIENT

0	0	0	0	0	0	1	1
---	---	---	---	---	---	---	---

**SUBTRACT DIVISOR FROM
 DIVIDEND ; SEND 1 TO
 QUOTIENT REGISTER ;
 SHIFT DIVISOR RIGHT**

50. The quotient is 110 (decimal 84 divided by 14 produces 6).

DIVIDEND

1	0	1	1	0	0	0	0
---	---	---	---	---	---	---	---

DIVISOR

0	0	1	1	0	0	0	0
---	---	---	---	---	---	---	---

QUOTIENT

0	0	0	0	0	0	1	1
---	---	---	---	---	---	---	---

COMPARISON :
 00001110 > 00000000

DIVIDEND

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

DIVISOR

0	0	0	0	1	1	1	0
---	---	---	---	---	---	---	---

QUOTIENT

0	0	0	0	0	1	1	0
---	---	---	---	---	---	---	---

**SEND 0 TO QUOTIENT ;
 STOP OPERATION**



ONE OF THE

BELL & HOWELL SCHOOLS

1. A - THE CLOCK PULSE GENERATOR IN A DIGITAL SYSTEM -- GENERATES THE BASIC TIMING SIGNALS.

Pulse generators in a digital system develop the timing pulses used to synchronize, or time, the various circuits in the system. These timing pulses are usually called clock pulses. The pulse generator which develops the timing pulses is usually called the clock pulse generator, clock generator, or simply the CLOCK.

2. D - IN DIGITAL SYSTEMS, THE CLOCK PULSES SYNCHRONIZE THE LOGIC CIRCUITS SO THEY PERFORM THEIR LOGICAL OPERATIONS -- ONLY WHEN THE CLOCK PULSES ARE APPLIED.

To synchronize the operation of the various circuits in a digital system, the circuits are arranged so they operate only when a clock pulse is applied.

3. A - IN THE RETIMING AND RESHAPING AMPLIFIER OF FIGURE 10, THE WIDTH OF THE OUTPUT PULSE IS DETERMINED BY THE -- WIDTH OF THE CLOCK PULSE.

The OR gate applies its input signal to the AND gate. When the clock pulse and the signals from the OR gate are both present, the AND gate delivers a signal to the pulse amplifier. The output signal from the pulse amplifier is applied back to the OR gate, where it combines with the input signal. The duration or width of the output pulse is determined by the width of the clock pulse. As soon as the clock pulse decreases to zero, the AND gate output signal no longer appears and the input to the pulse amplifier is removed. Thus, the time duration of the output signal is equal to the clock pulse. This is true even when the time duration of the input signal is stronger than the clock pulse. As soon as the clock pulse decreases to zero, no signal is delivered to the amplifier.

4. B - IF THE NUMBER STORED IN A COUNTER MADE UP OF FOUR CASCADED FLIP-FLOPS IS 1010, AFTER THE NEXT INPUT PULSE THE STORED NUMBER WILL BE -- 1011.

1010 is a Binary Coded Decimal (BCD) of the decimal number 10. The next input pulse would be a decimal 11 or a BCD of 1011.

5. A - CASCADED 6 FLIP-FLOPS PROVIDES -- 64 DIFFERENT STATES.

To determine the number of states a known number of cascaded flip-flops contains, the number 2^n is used, where n is the number of cascaded flip-flops. Since we have 6 cascaded flip-flops, our number becomes 2^6 which is 64.

6. B - A DEVICE USED FOR STORING ONE WORD IN A DIGITAL SYSTEM IS CALLED -- A REGISTER.

A clock generates the necessary timing signals; a gate controls or channels signals; an amplifier increases an input's voltage or current.

7. D - IN A SHIFT REGISTER, A SHIFT ONE PLACE TO THE LEFT -- DOUBLES THE VALUE OF THE STORED NUMBER.

Besides storing information, shift registers have provisions for shifting a number to the right or left with respect to the decimal point. For example, if a shift register is storing the binary number 1000.00 (decimal 8.0) and this number is shifted one place to the left, the binary number 10000.0 (decimal 16) results. Thus, a shift left results in doubling the value of the stored number.

8. D - DELAY LINES OR OTHER TEMPORARY STORAGE DEVICES MAY PRECEDE THE FLIP-FLOP INPUTS IN A STATIC SHIFT REGISTER TO -- OVERCOME RACING.

Racing is the problem which is created when a high speed flip-flop changes states during an input pulse. This results in an erroneous flip-flop state change. One way to overcome racing is to delay the flip-flop inputs by placing the contents of each flip-flop in a temporary storage device such as a delay line or another flip-flop. At some time after the shift pulse has ended, the stored information is transferred to the next flip-flop in the register.

9. A - IN A DYNAMIC SHIFT REGISTER, IF THE TOTAL DELAY TIME IS EQUAL TO ONE WORD-TIME, PLUS ONE BIT-TIME, THIS -- SHIFTS THE STORED WORD TO THE LEFT.

Shifting data in the dynamic register is somewhat different than shifting data in a static register. In the dynamic register, a word is stored in a dynamic device and the shift is accomplished by delaying or advancing a word in relation to the other words in the system. A one word-time minus one bit time total delay shifts the word to the right; a total delay of one word-time plus one bit-time shifts the word to the left.

10. D - BINARY MULTIPLICATIONS USING THE SHIFTING TECHNIQUE -- IS THE FASTEST AVAILABLE TECHNIQUE.

By shifting a binary number one place to the right, its value is halved. Therefore, by using shifting techniques, division of a binary number is made extremely simple and fast.

QUESTIONS

IMPORTANT—These instructions **MUST** be accurately followed to avoid loss, or errors in grading.

Indicate your answer on this sheet by filling in the box for the most correct answer to each question, as illustrated in the example below.

When all questions have been answered, place the answer card in the proper position to line up the boxes on the card with the boxes on the sheet.

Next, copy the complete lesson code into the space provided on the card, and fill in the answer boxes to correspond with those previously filled in on this sheet.

Before mailing, be certain your correct student number, name and address appear on the card.

LESSON CODE
5245A

Example: A square has

- | | |
|---------------------------------------|-----------------------------------|
| A ☐ | (A) three sides. (B) eight sides. |
| B ☐ | |
| C ☒ | (C) four sides. (D) ten sides. |
| D ☐ | |

1. A ☒ The clock pulse generator in a digital system
 B ☐
 C ☐ (A) generates the basic timing signals. (B) feeds information into and out of the system. (C) stores
 D ☐ information. (D) performs arithmetic operations.

2. A ☐ In digital systems, the clock pulses synchronize the operation of the logic circuits so that they perform
 B ☐ their logical operations
 C ☐ (A) while the signals are changing. (B) when no clock pulses are applied. (C) at random. (D) only when
 D ☒ the clock pulses are applied.

3. A ☒ In the retiming and reshaping amplifier of Figure 10, the width of the output pulse is determined by the
 B ☐ (A) width of the clock pulse. (B) amplitude of the clock pulse. (C) amplitude of the input pulse. (D) width
 C ☐ of the input pulse.
 D ☐

4. A ☐ If the number stored in a counter made up of four cascaded flip-flops is 1010, after the next input pulse
 B ☒ the stored number will be
 C ☐ (A) 1010 (B) 1011 (C) 1000 (D) 1111
 D ☐

5. A ☒ Cascading 6 flip-flops provides
 B ☐ (A) 64 different states. (B) 16 different states. (C) 32 different states. (D) 8 different states.
 C ☐
 D ☐

6. A ☐ A device used for storing one word in a digital system is called
 B ☒ (A) a clock. (B) a register. (C) a gate. (D) an amplifier.
 C ☐
 D ☐

7. A ☐ In a shift register, a shift one place to the left
 B ☐ (A) does not change the value of the stored number. (B) causes the value of the stored number to
 C ☐ decrease. (C) triples the value of the stored number. (D) doubles the value of the stored number.
 D ☒

8. A ☐ Delay lines or other temporary storage devices may precede the flip-flop inputs in a static shift register to
 B ☐ (A) decrease the storage capacity of the register. (B) speed up the shift register. (C) increase the storage
 C ☐ capacity of the register. (D) overcome racing.
 D ☐

9. A ☒ In a dynamic shift register, if the total delay time is equal to one word-time PLUS one bit-time, this
 B ☐ (A) shifts the stored word to the left. (B) does not shift the word. (C) has no effect on the value of the
 C ☐ word. (D) shifts the stored word to the right.
 D ☐

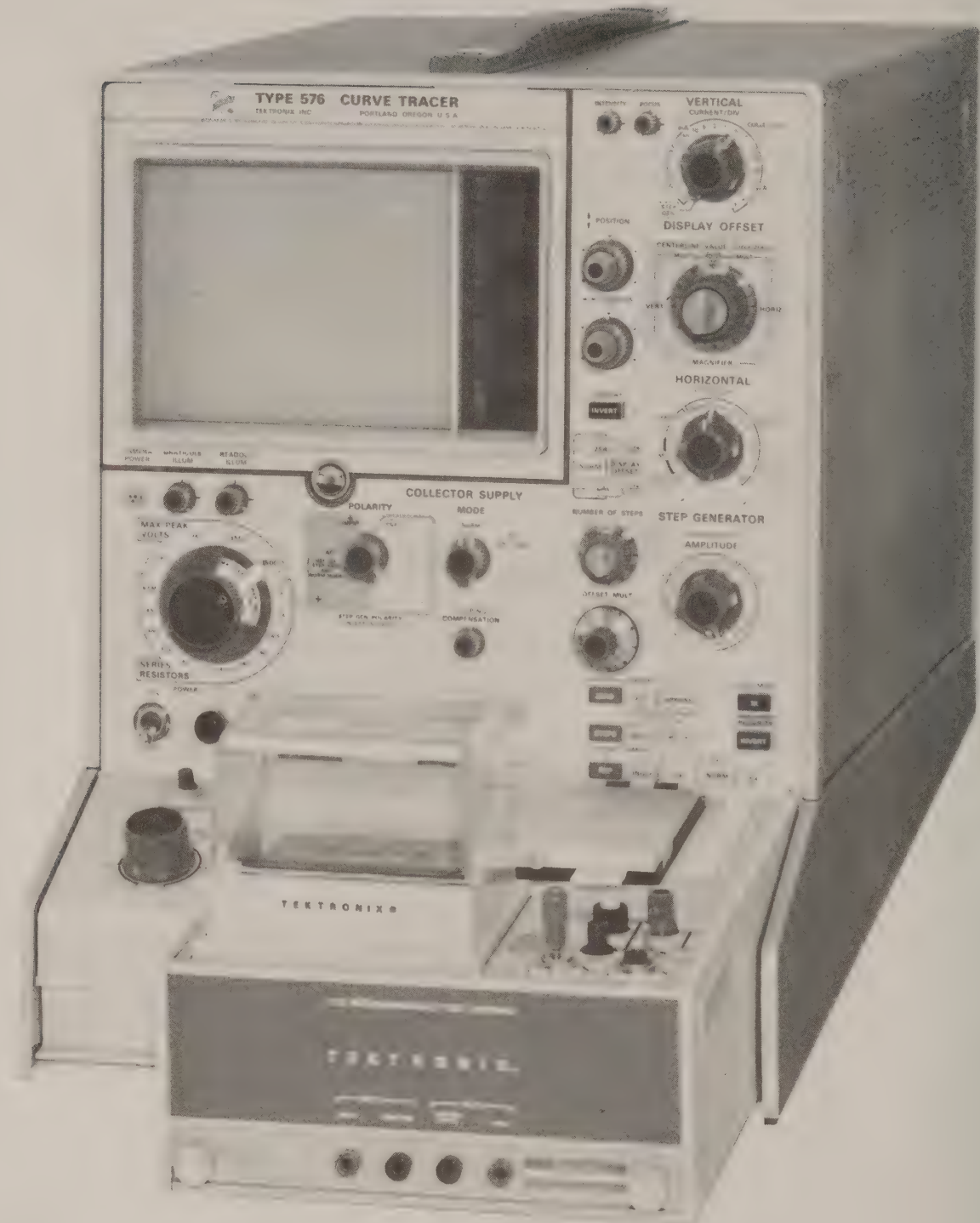
10. A ☐ Binary multiplication using the shifting technique
 B ☐ (A) requires fewer logic components than the other techniques. (B) is slower than the repeated addition
 C ☐ technique. (C) cannot be accomplished with fractional multipliers. (D) is faster than repeated addition.
 D ☒

DATA STORAGE CIRCUITS AND DEVICES

5250

© COPYRIGHT 1973
BELL & HOWELL SCHOOLS, INC.
ALL RIGHTS RESERVED





Curve tracers are capable of displaying one or more characteristic curves of diodes, transistors and IC's. The programmable test fixture attached to the curve tracer allows the operator to program up to eleven sequential tests on JFET's, transistors and diodes. This curve tracer is available with a long-persistence phosphor CRT.

Courtesy Tektronix, Inc.

CONTENTS

Memory Units	Page 5
Storage Characteristics	Page 6
Random Access Units	Page 7
Magnetic Memories	Page 9
Writing and Reading	Page 12
Core-to-Core Transfer	Page 14
Ferrite Apertures	Page 20
Thin-Film Memories	Page 20
Magnetic Drums and Discs	Page 23
Integrated Circuit Memories	Page 30

*I like the dreams of the future better
than the history of the past.*

—Patrick Henry

DATA STORAGE CIRCUITS AND DEVICES

A computing system requires both internal and external storage devices. A computer can automatically read from, erase, and write into the internal storage devices; however, input and output units must be utilized to read from or write into the external storage devices. Furthermore, data stored in some external devices, such as punched cards or punched or magnetic tapes, cannot be automatically erased by the computer. Although there are many types of storage units within a computer, 1-bit storage, buffer registers, arithmetic registers, etc., the term inner memory usually refers to the **MAIN MEMORY UNIT** of a computer.

The two main functions of a memory unit are: the storage of information used in arithmetic and data processing operations, and the storage of instructions used by the control unit. Most of the data are from the input unit; however, the memory unit must also store data from within the computer itself. The data consist of modified instructions and the intermediate and final results of arithmetic and data processing operations.

Each element of a memory unit capable of storing a logical digit is called a MEMORY CELL. A flip-flop circuit may form a memory cell. Many other devices also may be used. Heat, light, sound, electricity, and electromagnetism are all forms of energy used to store information. The most common storage devices in use today for large-capacity memories are magnetic devices in the form of tapes, drums, and discs. Smaller computers and desk-top calculators are taking greater advantage of memories formed on the basis of microminiaturization and integrated circuit technology. The ultimate goal of large scale integration (LSI) research and applications being "pushed" by the IC manufacturers is to make IC memories cost-competitive with the large-capacity drum, tape, and disc memory, as well as the already-miniaturized machine-wired **FERRITE CORE MEMORY**. It seems that IC's eventually will form the predominant bulk of all computer memories.

Magnetic cores are especially suited for storing and processing binary data. These magnetic cores can be magnetized in either of the two directions of magnetization (or two magnetic states) by a suitable driving current. The two magnetic states represent the two bits (0 and 1) of the binary number system. Only a small amount of current is required to switch a magnetic core from one state to the other. Much emphasis has been placed on making magnetic cores smaller and smaller. Currently, magnetic cores are no larger than a pin head. In the quest for smaller memory units, other magnetic memory cells have also been developed. One of the more common types is the **THIN-FILM MEMORY**, which is available in both ordinary and "supercooled" forms.

In this lesson we examine the structure, characteristics, and operating physics of the various storage devices. We also show how these storage devices are “tied together” to form large arrays of hundreds of thousands (even millions) of memory cells.

MEMORY UNITS

The bits of digital information (each composed of a logical 0 or a logical 1) are grouped to form words. Each word is stored in a particular location in a memory unit. Each location is assigned an **ADDRESS**, usually in the form of a binary number. For example, if a memory unit has 16 word locations, the addresses of these locations might range from 0000 (decimal 0) to 1111 (decimal 15), or any other convenient range of 16 values. If a system must reach a particular address by “sequencing” its way through all intermediate addresses first, it is said to be a **CYCLIC STORAGE** system. If a system goes directly to any desired memory address, it is known as a **RANDOM ACCESS STORAGE** or **RANDOM ACCESS MEMORY** system (the latter is called **RAM** by the semiconductor industry for IC memories). The latter can perform its operations much more quickly than the former. In some storage devices, the read-out operation destroys the stored information. In order to retain such information after read-out, the system must rewrite the same information at the same address after the read-out procedure is completed. This is an inefficient process. In a **NON-DESTRUCTIVE READ-OUT** unit, the read-out operation leaves the memory unit in its original state.

The type of storage device used to implement a memory unit of the required capacity is chosen through a compromise between two factors: **ACCESS TIME** and cost. Access time is the time required to transfer data between the memory unit and the other units in the computer. In comparing the cost and capacity of different types of memory units, the cost is usually expressed in terms of cost per bit of storage capacity. For example, if the cost per bit of a certain storage device is 25 cents, then a unit with a storage capacity of 100,000 bits would cost \$25,000.

Fast storage devices are usually more expensive than slow storage devices. Thus, extremely fast devices are usually used only when speed is the most critical design characteristic. When possible, large-capacity units are made up of slower, less expensive devices. Flip-flops are extremely fast, but relatively expensive devices, especially when they are discrete circuits; therefore, they are most commonly used for high speed registers and accumulators in arithmetic units. Although the cost per bit is high, the total register capacity is not very great. Conversely, the main memory is made up of less expensive storage devices (in terms of cost per bit). The main memory has a longer access time and a lower cost per bit than the registers.

The access time of the register storage units in a digital computer is negligible, because it contains flip-flops. Thus, this type of storage is said to be zero access. The access time of the main memory unit is not negligible; however, it still compares favorably with the operating speed of the other computer units.

In some computers the main memory unit is made up of different sections using different types of storage devices. The access times of some of the sections may be longer than others; however, they are all short enough so they do not seriously slow down the overall operation of the computer.

In addition to the main memory unit and the relatively small register units, a computing system may also have other fairly large storage units called **AUXILIARY STORAGE UNITS**. Auxiliary units use even slower storage devices, at an even lower cost per bit. Together, all of the auxiliary storage units may represent a larger capacity than the main memory unit. Data usually cannot be transferred directly between the auxiliary storage units and the main unit. Instead, blocks of data from the auxiliary storage unit are first transferred to a buffer register, then to the main memory. To insert data into an auxiliary unit, the data are first transferred from the main memory to the buffer, then from the buffer to the auxiliary unit.

Buffers are also a form of temporary storage. They are used to compensate for speed differences between various parts of a digital system. Buffer units may also be required between the machine units and the external storage units (input-output buffers).

The computer continues to perform arithmetic and data processing operations while data are being transferred between buffer storage and secondary or external storage.

Storage Characteristics

Three important storage characteristics (capacity, cost per bit, and access time) have already been described. A more detailed description of the term access time is required for a thorough understanding of memory units. The access time of a cyclic memory unit is made up of the **LATENCY TIME** plus the word time. The latency time is the time required to select an address requested by the control unit. For example, in a magnetic tape storage system the latency time is the time spent waiting for the desired address to appear under the read-write heads. Once read-out or write-in begins, 1 word time is required to read-out or write-in the desired word.

Random access storage units are non-cyclic. The access time of random access memories does not depend upon the address selected, or upon the previous address position in the memory.

Storage systems can also be classified as **STATIC STORAGE** or **DYNAMIC STORAGE**. In a static storage system, the data are said to exist in space, since the location of the memory cells is fixed in space, and a particular address is selected by selecting a few desired memory cells out of many. Flip-flop registers, RAM's, and magnetic core storage units are typical examples of static storage units.

In a dynamic storage system, data continuously change position as in a delay line or magnetic tape storage unit. In such a storage system, data are said to exist in time rather than in space. Existing in time means that the time relation among the bits inserted in the memory remains fixed, and a particular address is selected by determining when it passes the read-write heads.

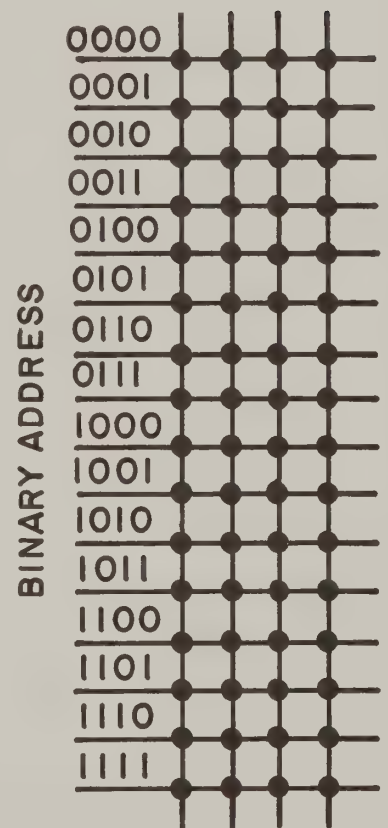
Some storage devices require the continuous application of electrical energy to retain the stored information; therefore, they are called **VOLATILE STORAGE** devices. The removal or loss of power destroys or changes the contents of volatile storage devices. In volatile storage devices, the stored information may have to be continuously regenerated by reading it out, amplifying it, and writing it back into its previous address. Many types of dynamic storage devices (such as delay lines) are volatile storage devices. If the removal or loss of power does not destroy or change the contents of a storage unit, it is a **NON-VOLATILE** or **PERMANENT STORAGE** unit.

In some storage devices, read-out destroys the stored information. This is called **DESTRUCTIVE READ-OUT**. To retain information in the memory unit after read-out, the information must be rewritten into the same address from which it was read. In a non-destructive read-out, the read-out does not destroy the stored data.

Random Access Units

Bistable memory cells are arranged in rectangular arrays of rows and columns called a **MATRIX** (as shown in Figure 1). The large black dots represent memory cells. Magnetic cores can be placed in such an arrangement to provide random access. This is an example of static storage.

Since random access memory units are made up of bistable memory cells fixed in space, the data are stored in space and the address selection method is called **Spatial Selection**, or simply **S-SELECTION**.



**THE MATRIX
ARRANGEMENT
FOR A RANDOM
ACCESS MEMORY**

Figure
1

The memory cells in a random access memory are divided among a certain number of locations; each location can store a computer word. The number of memory cells in any word location must equal the number of bits in the word. Each word location is assigned an address in the form of a binary number. Usually, the number of locations is equal to the maximum number of binary addresses which can be created from a certain number of bits. For example, a 2-bit address can only provide 2^2 (or 4) addresses; a 3-bit address can provide 2^3 (or 8) addresses, and so on. Large memories may require addresses composed of 14 to 16 bits (16,384 to 65,536 addresses). The matrix in Figure 1 shows 16 words of 4 bits each (a total of 64 bits storage). The word size could be extended simply by adding more memory cells along each horizontal row. This would have no effect on the addresses. However, a 4-bit address code can only provide 2^4 (or 16) addresses (the number shown in the figure). To increase the number of word locations, the address code size must be increased. Using a 5-bit address code would allow 32 word locations, thus doubling the memory size from the 4-bit code.

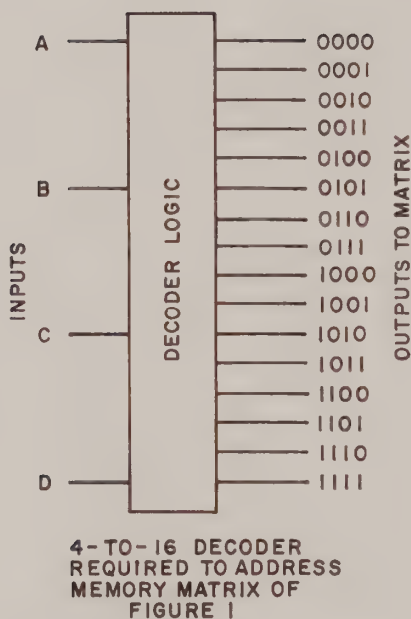


Figure
2

Exceptionally large memories may use a three-dimensional array or matrix of memory cells. Then, two sets of addresses could be used. The first address would select a "layer" of cells within the cube; the second address would select the row (as in Figure 1).

The memory arrangements which have been described are known as **WORD-ORGANIZED MEMORIES**. That is, one address code permits the selection of a complete word in the memory. In order to write a word into an address or read a word out of the address, a decoder must be used to enable that particular location. Figure 1 requires a 4 line binary to 1-of-16-output decoder. This decoder is shown in Figure 2.

Simpler decoders can be used if each memory cell is made to take part in the address selection process. Figure 3 shows a 64-bit memory arranged for this purpose. There are four 16-bit "layers" or planes. Although shown "spread out," the four planes would actually be stacked one behind the other to form a three-dimensional memory. Each plane contains one bit of the 4-bit words held in the memory; therefore, this memory is identical to Figures 1 and 2 in the sense that it will hold sixteen 4-bit words. However, with this arrangement, the sixteen words are addressed using two sets of 2-bit address codes. This permits the use of two 2-line binary to 1-of-4 decoders. This arrangement is considerably less complicated than the decoder in Figure 2. The matrix of Figure 1 and the decoder of Figure 2 can be combined with other read and write lines to form the circuit shown in Figure 3.

As an example of the operation of the memory arrangement in Figure 3, assume that we wish to access the word indicated by bits $A_1-A_2-A_3-A_4$. This is accomplished by activating the $X = 11$ selection line and the $Y = 10$

selection line. The $X = 11$ line provides a partial activation to all the bits along the horizontal line at the bottom of the array. However, for a bit to be completely activated, it requires a simultaneous X and Y signal. If you trace the $Y = 10$ line throughout the memory, you will see that it intersects only the A_1 , A_2 , A_3 , and A_4 bits in the bottom horizontal line. For $X = 11$ and $Y = 10$, the A_1 , A_2 , A_3 , and A_4 bits are the only bits with simultaneous X and Y signals.

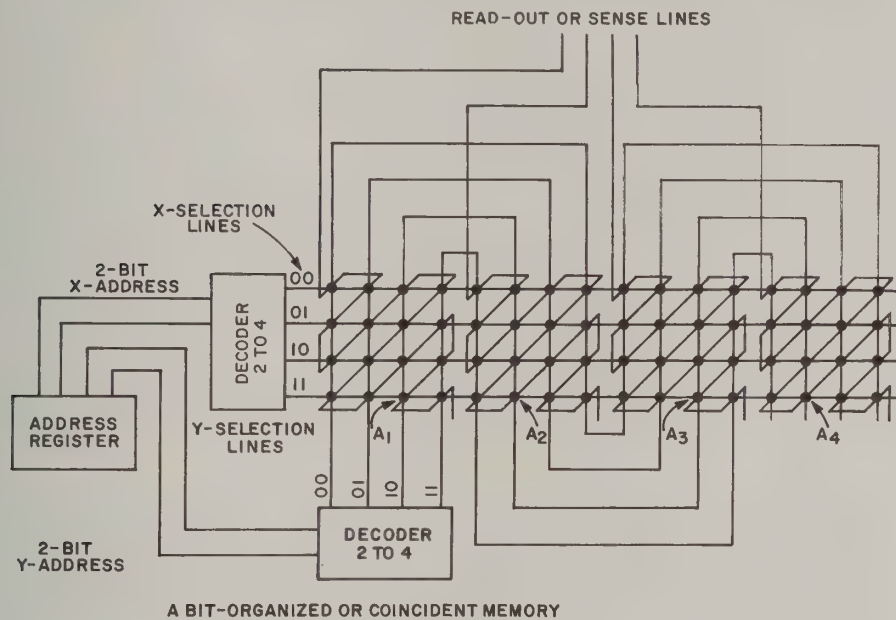


Figure 3

Memory units arranged as in Figure 3 are called **BIT-ORGANIZED MEMORIES, COINCIDENT-CURRENT MEMORIES**, or simply **COINCIDENT MEMORIES**. At least two selection signals are required to access a particular address in a coincident-current memory unit.

MAGNETIC MEMORIES

To build large memory units capable of storing millions of bits, inexpensive memory cells are required. Magnetic devices, such as cores, ferrite apertures, drums, and discs, are among the more inexpensive storage devices which provide the required logic for each memory cell. Thin-film memory devices are somewhat more expensive; however, they are used when high speed is required and limited space is available.

Magnetic cores are most commonly used to provide efficient storage units in digital systems. Magnetic cores are also used in other types of digital

circuits. When a core is switched from one magnetic state to another, the flux which makes up the magnetic field of the core reverses direction. This change in flux can induce a voltage across a coil wound on the core in the same way that voltages are induced across transformer windings. The induced voltage is the output of the core. The output voltage may be many times greater than the controlling input voltage. Thus, the magnetic core may act as a pulse amplifier. Other digital circuits such as gates, registers, and counters are usually formed by interconnecting magnetic cores in combination with diodes or transistors.

Two types of materials are used in most digital magnetic applications. These two types of material are ferromagnetic metals and ferromagnetic ceramics (ferrites). The term "ceramic" refers to a hardened clay-like material and a ferrite is a type of chemical compound containing iron. The prefix "ferro" indicates an iron compound, the most common natural ferromagnetic material. The term ferromagnetic has now been broadened to include any material which presents a small opposition to flux lines. Ferromagnetic materials either have a high permeability or a low reluctance. Besides iron, the elements cobalt and nickel are ferromagnetic materials. Alloys (mixtures of substances at least one of which is a metal) of ferromagnetic materials are also ferromagnetic. Alloys of ferromagnetic materials and non-ferromagnetic materials may also be ferromagnetic.

Nickel-iron alloys (called permalloy) are commonly used for digital magnetic devices. To obtain the desired characteristics, it is common to also include a small percentage of the element molybdenum in these alloys to form molybdenum permalloy. Numbers preceding the term permalloy are often used to indicate the percentage of each metallic element in the alloy. For example, 65 permalloy is 65% nickel and the balance of the material, 35%, is iron; 4-79 permalloy is 4% molybdenum, 79% nickel, and the remaining 17% is iron; and 2-81 permalloy is 2% molybdenum, 81% nickel and 17% iron.

Magnetic cores are shaped like doughnuts (called toroids). The most common method of construction for ferrite cores is to mold a powdered ferrite together with a ceramic binder. The mold is heated and the material is allowed to crystallize. Molded ferrite cores range in size from about 0.03 inch to 0.25 inch in the outside diameter. The magnetic alloy cores are made of several turns of a very thin ribbon of the metal, wound in tape-like fashion around a ceramic, plastic, or stainless steel bobbin. These cores range in size from about $\frac{1}{8}$ -inch to $\frac{1}{2}$ -inch outside diameter. The smaller cores allow higher speed switching.

As shown in Figure 4, magnetic cores may be magnetized in either a clockwise (logic 0) or a counterclockwise (logic 1) direction by suitable currents in the X and Y wires passing through them. The current in the X and Y leads must go through the core in the same direction (or the same polarity). The Y current cannot be $+\frac{1}{2}I_m$ (down) while X is $-\frac{1}{2}I_m$ (right).

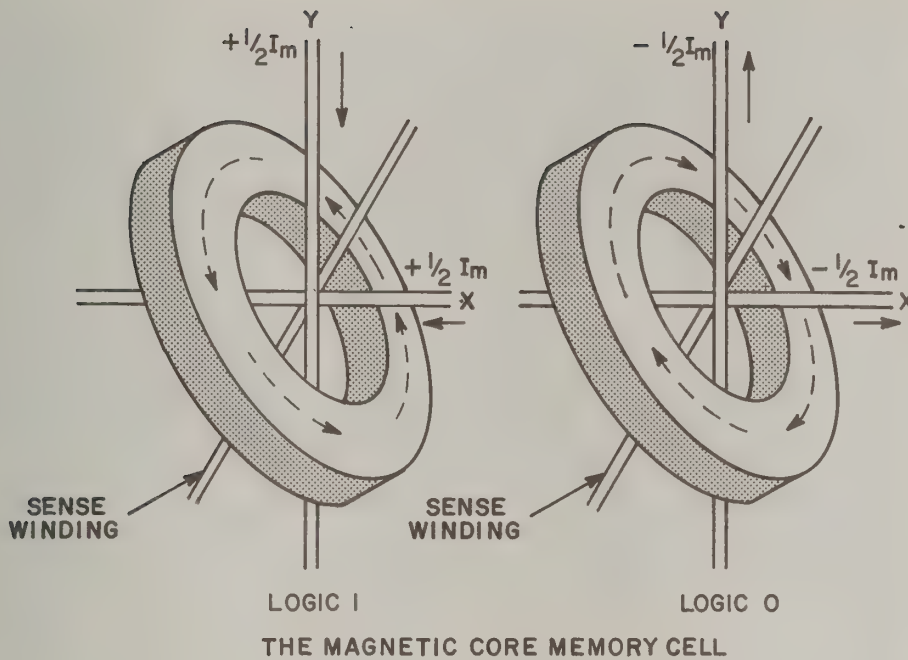
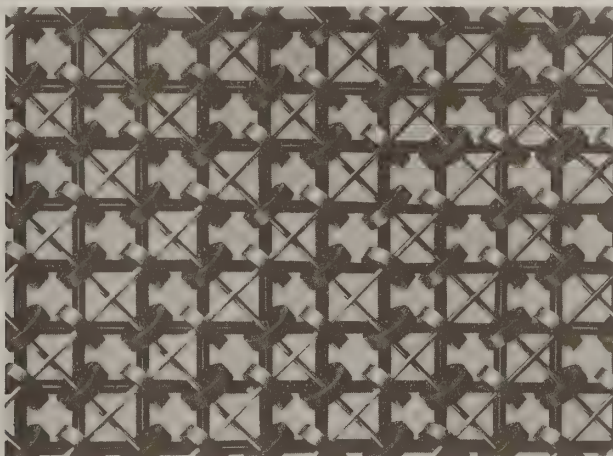


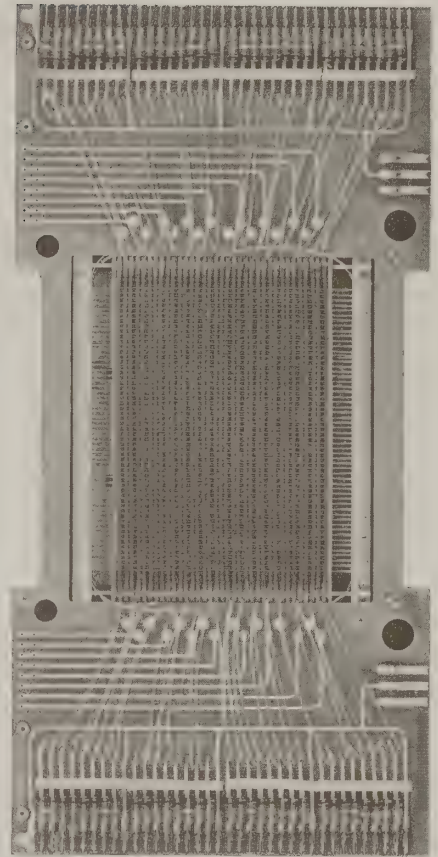
Figure 4

Such current will not change core states. In addition to providing the driving current to magnetize the cores, the X and Y wires provide structural support for the cores in a memory matrix. A typical ferrite core plane for random access is shown in Figure 5. These ferrite cores have an outside diameter of 0.03 inch; therefore, they are not visible in the figure. Figure 6 shows a magnified section of a core plane. The cores, X and Y wires, and the **SENSE WIRES** are readily seen. A three-dimensional stack of core planes is shown in Figure 7.



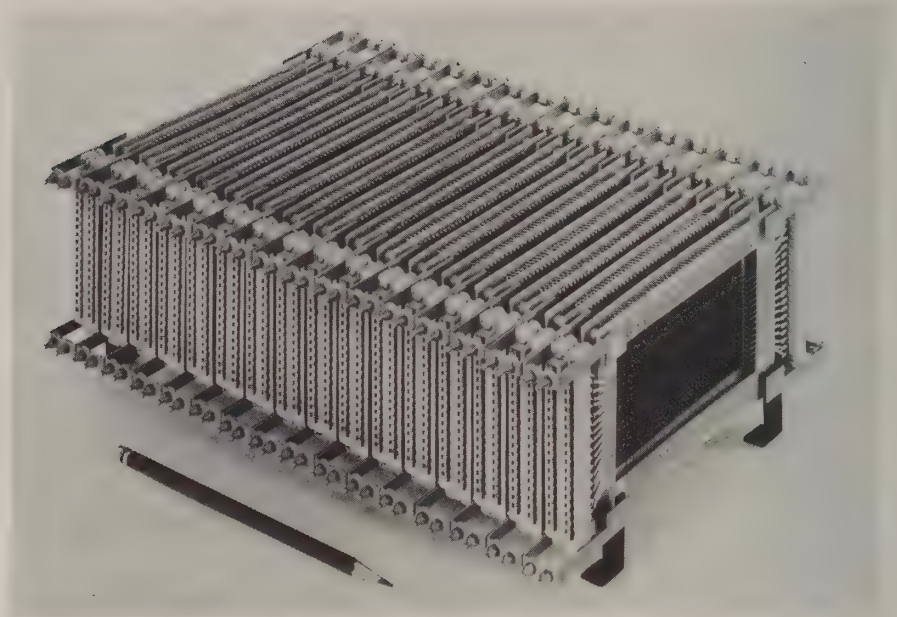
Magnification of a Portion of a Ferrite Core Memory Plane
 Courtesy Ampex Computer Prod. Co.

Figure 6



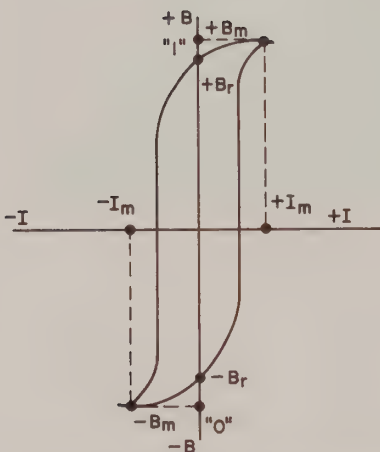
Typical Ferrite Core Memory Plane
 Courtesy Ampex Computer Prod. Co.

Figure
 5



Complete Memory Unit of Stacked Core Planes
Courtesy Ampex Computer Prod. Co.

Figure 7



HYSTERESIS LOOP FOR A MAGNETIC CORE
AND THE LOGIC STATES "0" AND "1"

Figure
8

Sense wires are used to read the state of the core. To accomplish this, negative current pulses are applied to the X and Y wires (Figure 4). This causes cores in the 1 state to switch to the 0 state and leaves cores in the 0 state in their original condition. When the cores in the 1 state switch, a current pulse is induced in the associated sense wire. This is called a destructive form of read-out.

Writing and Reading

In the materials used for magnetic storage devices, the magnetic flux density (denoted by B) is not proportional to the intensity of the applied current. In fact, the relationship between B and I is different for different directions of the magnetizing current. This is the property of magnetic hysteresis, resulting in the characteristic curve shown in Figure 8.

Once a positive current pulse of sufficient amplitude to saturate the core is applied, a **REMANENT FLUX DENSITY** continues circulating even after the pulse is removed. This is shown by the flux density level, B_r , when the magnetizing current is zero. This level of flux density represents the 1 state (where the core is set). A similar condition occurs if a negative current pulse causes core saturation. However, this time the remaining flux density level represents a circulation in the opposite direction. This is indicated by $-B_r$ in the figure, and represents the 0 state (where the core is reset).

The process of inserting a bit of datum (0 or 1) into a core is called writing, or write-in. The process of sensing the state of a core is called reading, or read-out. Before a datum bit is inserted into a core, the core is usually cleared to the 0 state. This may be done during the read-out process. If a 0 is to be inserted into the core, it is not necessary to switch the core since it is already in its 0 state. However, if a 1 is to be inserted into the core, the core must be switched from its 0 state to its 1 state. This is done by a pulse of magnetizing current called a write pulse.

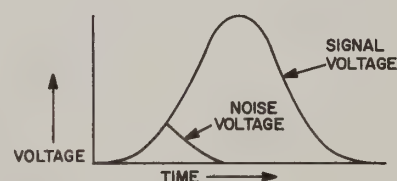
The write pulse amplitude must be equal to a magnitude of at least I_m in order to reverse the core state (as shown in Figure 8). A smaller pulse magnitude will simply create a temporary change in flux direction, allowing it to fall back to the original remanent level when the pulse is removed. Since two write wires are used in Figure 4, the current load is split between them. Each write wire carries a pulse with an amplitude equal to at least $\frac{1}{2}I_m$. Using the appropriate decoders, addressing the proper core and writing a 1 onto the core can be accomplished in the same operation since only the core with both X and Y driving currents of sufficient magnitudes will be switched. Other cores either experience only the X-current or the Y-current. This only causes a momentary change in state, then the core returns to its original 0 value.

When a memory circuit has information written into it, some cores will be in a 1 state; others, in a 0 state. These states can be sensed (or "read") by applying simultaneous X- and Y-current pulses (read pulses) of amplitude $-\frac{1}{2}I_m$ to the desired cores. A set of read pulses will drive a core already in the 1 state into the 0 state; however, it will not have any significant effect on a core already in the 0 state (as shown in Figure 8). The 0-state core simply changes from $-B_r$ to $-B_m$, then back to $-B_r$ when the read pulse is removed.

The changes in core flux density when the read pulse is applied induce a current pulse in the sense wire shown in Figure 4. A core originally in the 1 state produces a read-out pulse of magnitude I_m in the sense wire. A core originally in the 0 state produces a very low-magnitude current pulse. This pulse is considered as "noise," since many other low-level signals occur in the system. The relative amplitudes of signal and noise voltages are shown in Figure 9. Signal-to-noise ratios of 20 to 1 or greater are commonly obtained with commercially available cores.

The read pulse always leaves the core in the 0 state. When necessary, the data read-out of a core is written back into the core to obtain, in effect, a non-destructive read-out.

In magnetic core circuits, data cannot be read out at the same time new data are inserted into the cores. Read-out must occur sometime after write-



COMPARISON OF SIGNAL AND NOISE VOLTAGES OBTAINED FROM MAGNETIC CORE READ-OUT

Figure
9

in. The basic clock cycle is therefore made up of at least two clock pulses. Read-out occurs during one clock pulse and write-in occurs during the other clock pulse. The clock pulses are phased to occur at the proper times during a basic clock cycle. The clock phases may be labeled t_1 , t_2 , etc. The subscripts indicate the order in which the pulses occur during the basic clock cycle. One phase of the basic clock cycle usually provides the read pulses for magnetic core circuits. Therefore, read pulses are often referred to as clock or timing pulses.

The power available from the output of a magnetic core circuit often depends on the power available from the clock source. The clock pulses used in magnetic core circuits are often applied to power amplifiers before being applied to the magnetic core circuits themselves. Using high power clock pulses insures that the outputs of a magnetic core circuit have enough power to switch the required number of circuits.

Core-to-Core Transfer

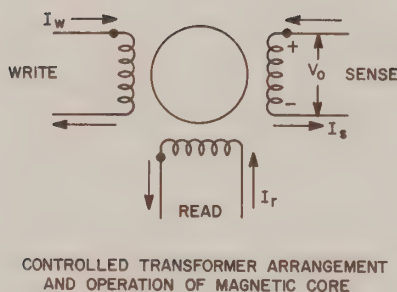


Figure
10

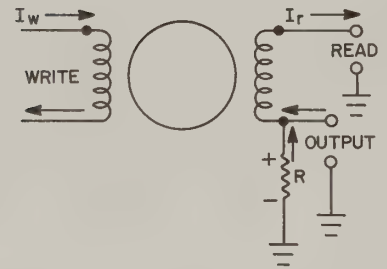
A magnetic core may be used either as a controllable transformer or as a variable impedance. When a core is used as a controllable transformer, at least three windings are required: one each for writing, reading and sensing (as shown in Figure 10). Current in the write winding switches the core to the 1 state, inserting a bit of datum. Current in the read winding clears the core to the 0 state, inducing a voltage in the sense winding. The voltage induced in the sense winding before the application of the read current is the output which indicates whether the core was in the 0 or 1 state. The sense winding senses the change of flux when current is applied to the read winding. (The cores in Figure 6 are used as controllable transformers. This is why there are double X- and Y-wires.)

To examine the operation of the circuit in Figure 10, assume that the core is set to its 1 state by a current, I_w , in the write winding. Current I_r through the read winding causes the core to switch to its 0 state, thus producing a large change in flux. The large change in flux induces a voltage, V_o (of the polarity indicated in Figure 10), across the sense winding. If the output circuit is complete, electrons flow in the direction of I_s . Output voltage, V_o , corresponds to the signal voltage of Figure 9. If the core is not set by I_w , the application of I_r only causes a small change of flux. The small change of flux induces a small voltage across the sense winding. This corresponds to the noise voltage of Figure 9. If the core is in its 1 state, a large voltage of the indicated polarity appears across the sense winding; however, if the core is in its 0 state, very little voltage appears across the sense winding.

When the core is used as a variable impedance, only two windings are required: a write winding and a read winding (as shown in Figure 11). The

read winding also serves as the sense winding. Data are inserted into the core in the same way as when the core is used as a controllable transformer. Current in the write winding switches the core to the 1 state. The method of sensing the state of the core of Figure 11 is based on the impedance of the read winding during switching. The impedance of the winding may be large or small, depending upon the amount of flux change caused by I_r . Two impedance rules for sensing the state of a core are:

- (1) If the read current switches the core from the 1 state to the 0 state, there is a large change in flux. This large change in flux cuts the read winding, inducing a large counter emf. Because of the large counter emf, the winding exhibits a large impedance.
- (2) If the core is already in its 0 state, there is only a small change in flux. This small change in flux induces a small counter emf in the read winding. Thus, the winding presents a very low impedance.



CONTROLLED IMPEDANCE ARRANGEMENT
AND OPERATION OF MAGNETIC CORE

Figure
11

Figure 11 shows one way that the variable impedance of the read winding can be used to sense the state of a core. The read winding is connected in series with a resistance, R . If the core is in its 1 state when the read current, I_r , is applied, the read winding has a high impedance, and there is very little current through R . With very little current through R , the voltage across R is small. If the core is in its 0 state when the read current, I_r , is applied, the read winding has a low impedance, and there is a large current through R . With a large current through R , the voltage across R is large. Thus, if the core is in its 1 state, very little voltage appears across R ; however, if the core is in its 0 state, a large voltage of the polarity indicated appears across R .

In both the controllable transformer and the variable impedance methods, the read pulse clears the core. This is called destructive read-out since the binary bit stored in the core is destroyed (removed).

The output pulse from one core may be used to switch another core. The circuits that transfer data from one or more cores to one or more other cores are called **TRANSFER LOOPS**. A transfer loop usually includes the output windings of the transmitting cores, the input windings of the receiving cores, and one or more diodes. All of the control, storage, and logical operations of a digital system can be implemented by using three fundamental transfer loops. These three loops are called the **SINGLE-DIODE TRANSFER LOOP**, the **SPLIT-WINDING TRANSFER LOOP**, and the **INHIBIT TRANSFER LOOP**.

The **SINGLE-DIODE TRANSFER LOOP** provides an **UNCONDITIONAL TRANSFER** of data from one core to one or more other cores. An unconditional transfer means that the state of the transmitting core transfers to

the receiving core when any input resets the transmitting core (to the 0 state). No special conditions must be met to transfer data from one core to another. Figure 12A shows the schematic diagram of a single-diode transfer loop. In this circuit data are transferred from core A to core B. In single-diode transfer loops, the cores are used as controllable transformers. The diode in this loop allows electrons to flow in one direction only.

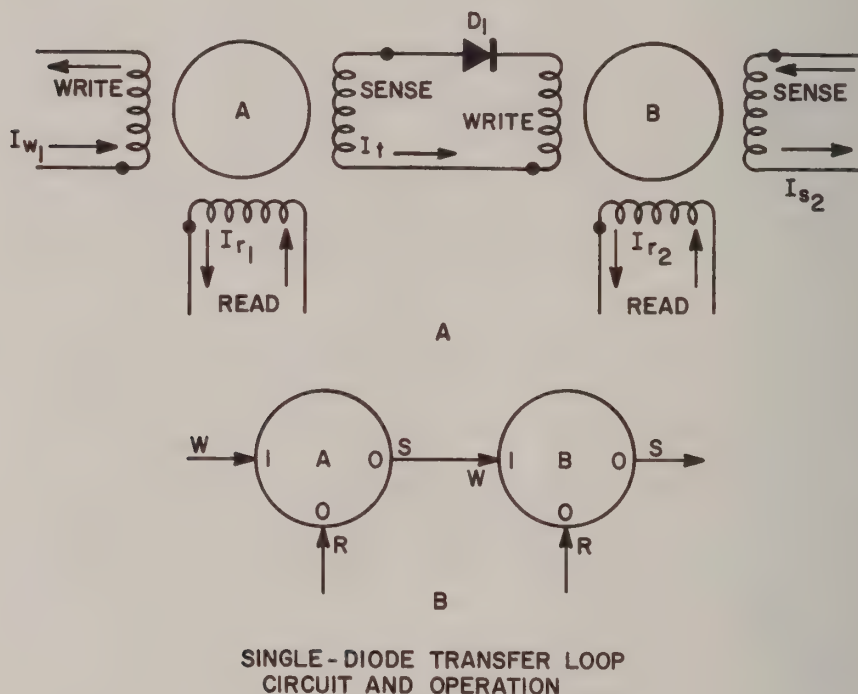


Figure 12

To examine the operation of the circuit, assume that core B is in the 0 state and core A is in either the 0 state or the 1 state. If you apply a read pulse to the read winding of core A, current I_{r1} in the winding causes a voltage to be induced across the sense winding of core A if the core is in the 1 state. This induced voltage produces transfer current I_t in the transfer loop. I_t in the write winding of core B thus sets core B. If core A is in the 0 state when the read pulse is applied, no voltage is induced across the sense winding, thus no transfer current occurs in the transfer loop; therefore, core B remains in the 0 state. A read pulse applied to core A transfers a 0 or 1 (whichever core A contains) to core B (B must be initially at 0). When core B is in the 0 state and a 0 is transferred from core A into core B, the states of the two cores do not change. Therefore, we can say that we have transferred a binary bit from A to B.

If you switch core A from the 0 state to the 1 state by applying a pulse to the write winding of core A, current I_{w1} occurs in the winding, and a

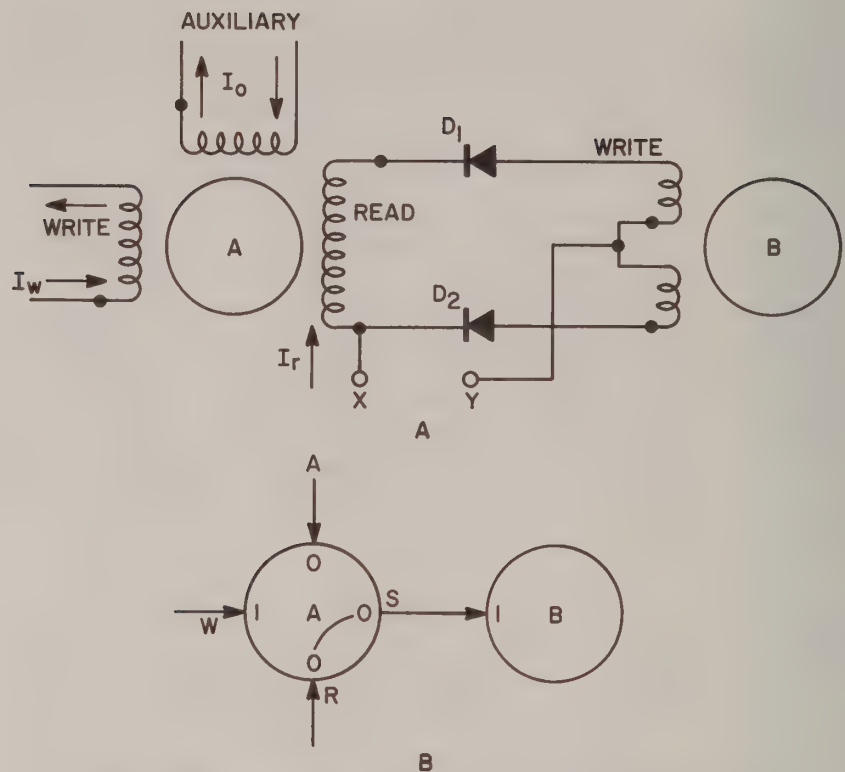
voltage is induced in the sense winding of core A. This induced voltage tries to produce current in the direction opposite to I_t . A current in this direction would clear core B. Diode D_1 is therefore placed in the transfer loop to block this current.

If you apply a read pulse to the read winding of core B when core B is in the 1 state, current I_{r2} occurs in the winding and a voltage is induced across the write winding of core B. This voltage produces a current in the same direction as I_t . A large current in this direction through the sense winding of core A would switch core A to its 1 state. D_1 presents a low value of impedance to the transfer current I_t , produced by a voltage across the sense winding of core A. It also presents a high value of impedance to a current in the same direction, produced by a voltage across the write windings of core B. Therefore, the value of current in the transfer loop when core B is switched from the 1 state to the 0 state is not great enough to set core A.

Although Figure 12 shows only one receiving core (core B), additional receiving cores may be switched by connecting their write windings in series in the transfer loop. The single-diode transfer loop allows a binary core to store a bit of datum indefinitely without power dissipation. It also allows unconditional transfer of that bit to one or more receiving cores when a read pulse is applied to the transmitting cores. Figure 12B shows the logical block diagram corresponding to the schematic diagram of Figure 12A.

The SPLIT-WINDING TRANSFER LOOP provides a CONDITIONAL TRANSFER of data from one core to one or more other cores. A conditional transfer means a special condition must be met to transfer data from one core to another. The state of the transmitting core transfers to the receiving cores only when a certain input resets the transmitting core. Figure 13A shows the schematic diagram of a split-winding transfer loop. The write winding of core B is split into two windings. Data are transferred from core A to core B. In the split-winding transfer loop, the cores are used as variable impedances. The diodes in the split-winding transfer loop serve the same purpose as the diode in the single-diode transfer loop.

To examine the operation of the circuit of Figure 13A, assume that both cores are in the 0 state. Read current I_r has the direction shown. There are two parallel paths for I_r from point X to point Y. One path is from point X up through the read winding of core A, through D_1 , down through the upper half of the write winding of core B, and back to point Y. The other path is from point X through D_2 , up through the lower half of the write winding of core B, and back to point Y. Since core A is in its 0 state, its read winding has a very low impedance. For all practical purposes, the two parallel paths for I_r have the same impedance; therefore, each path carries



SPLIT-WIRE TRANSFER LOOP CIRCUIT AND OPERATION

Figure 13

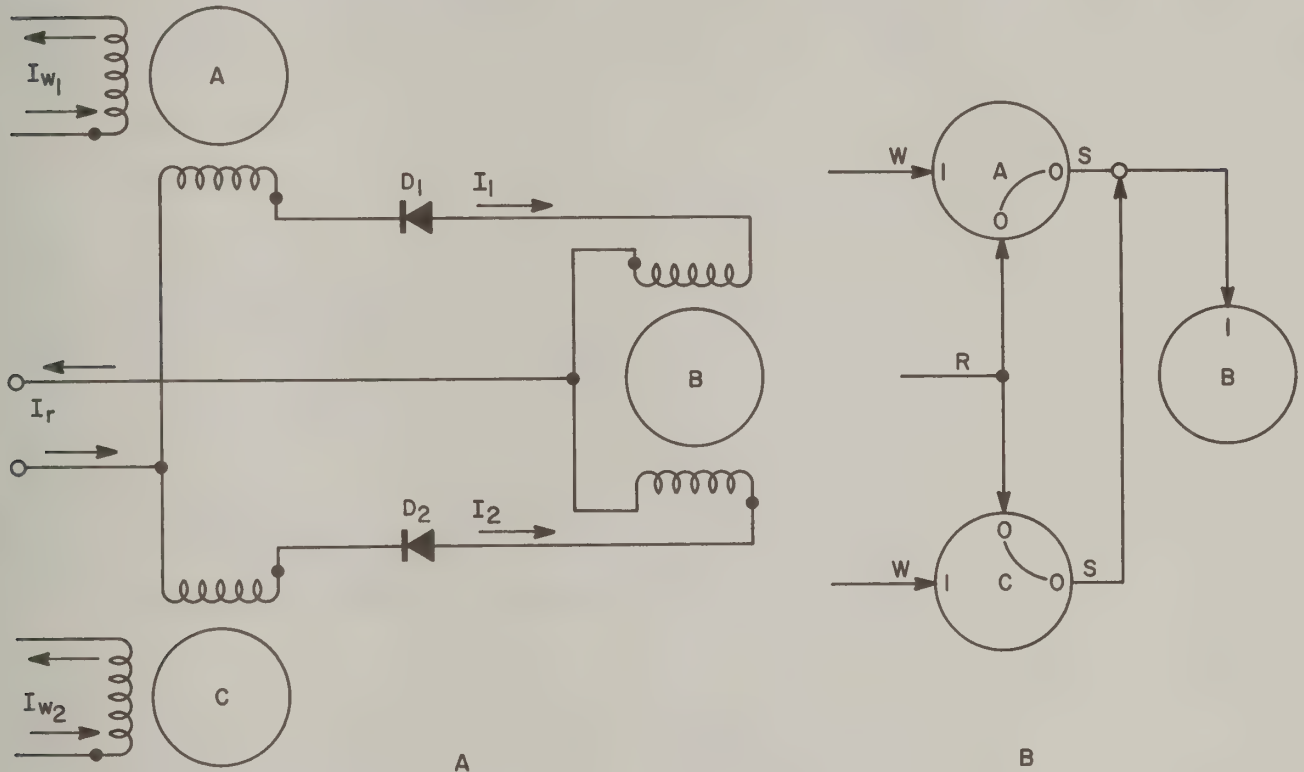
the same amount of current. In the write winding of core B, current is present in the upper and lower halves of the winding. The net flux is zero and core B remains in the 0 state.

Now, assume that core A is in the 1 state and core B is in the 0 state when the read pulse is applied. Since the impedance of the read winding is very large, there is much less current in the upper half of the write winding of core B than there is in the lower half of the winding. Thus, the resulting flux sets core B, and core A is reset.

Unlike the single-diode transfer loop, the split-winding transfer loop is a conditional transfer loop. Simply switching core A from the 1 state to the 0 state does not transfer the 1 to core B. In order to transfer a 1 from core A to core B, a read pulse must be applied to the read winding. Current I_o in the auxiliary winding of core A clears core A but does not transfer a 1 to core B.

Figure 13B shows the logical block diagram corresponding to the schematic diagram of Figure 13A. The curved line inside the circle, representing core

A, indicates that an output pulse is present only when input R is present. Only input R can transfer data from core A to core B, leaving A in the 0 state. Input A can reset core A but cannot transfer data from core A to core B. Thus, the curved line represents a conditional transfer, and is sometimes called an **EYEBROW SYMBOL** because of its physical appearance.



INHIBIT TRANSFER LOOP CIRCUIT AND OPERATION

Figure 14

Figure 14A shows the schematic diagram of an **INHIBIT TRANSFER LOOP** which also provides conditional transfer of data from one core to one or more other cores. The inhibit transfer loop is a special form of split-winding loop interconnecting cores A and C. A binary bit is transferred from core A to core B only if core C is in the 0 state and if B is initially in the 0 state. A 1 in core C inhibits the transfer of a 1 from core A to core B, because I_1 is equal to I_2 when the read pulse is applied, producing current I_r . Since I_1 is equal to I_2 , core B does not switch to the 1 state. This is also true when both core A and core C are in the 0 state.

When core A is in the 1 state and core C is in the 0 state, I_2 is greater than I_1 . The resulting magnetizing force sets core B. Also, if core A is in the 0

state and core C is in the 1 state, I_1 is greater than I_2 . The resulting magnetizing force clears core B. In each case, I_1 clears both core A and core C. Since core B is normally cleared by its own windings, the logical block diagram for the circuit (Figure 14B) need only show the inhibiting action of core C on the transfer of a 1 from core A to core B. The small circle on the input line to core B shows that an output from core C inhibits the transfer of data from core A to core B. Many logical operations difficult to implement with other transfer loops are very easy to implement with the inhibit transfer loop.

Ferrite Apertures

A solid sheet of ferrite material, in the form of a thin plate, with regularly spaced holes or apertures can be used in the place of a matrix of ferrite cores in order to form a memory unit. These plates are called FERRITE APERTURED PLATES. The material around each hole can be made to support a magnetic flux, circulating in either direction, just as the core does. The holes are written in and read out using conductors in the same way in which conductors are used in magnetic cores. Since the magnetizing force within a material is inversely proportional to the distance from the conductors, there is an insignificant interaction (noise) between neighboring holes as long as the driving currents are kept below some threshold value. This value is reduced as the holes are spaced closer together, thus placing a limit on the hole density.

Ferrite apertured plates have several advantages over magnetic cores. Since ferrite materials are poor electrical conductors, current windings can be placed directly on the plates in printed circuit fashion. Smaller driving currents can be used since the flux is passed through smaller volumes of ferrite. Apertures can be made much smaller than cores, because the cores become impossible to handle when they are made extremely small. A typical aperture array of 16 by 16 holes (256 bits) is less than one inch square.

Thin-Film Memories

Memory cells in a random access memory unit can be formed by depositing ultrathin films of a nickel-iron alloy on a thin glass substrate. For different memory units, the film thickness may range from 500 to 5000 Angstroms. For any given unit, the thickness must be controlled within a few hundred Angstroms. (One Angstrom is 0.000000004 inch.) Thin-film memory units have several advantages over conventional magnetic memories. They can generally be cycled in less time than for a magnetic core memory. Normally, magnetic cores are cycled over longer time periods because excessive excitation power is required at higher speeds.

The following Practice Exercise questions cover the subjects which you have just studied. They are:

MEMORY UNITS

STORAGE CHARACTERISTICS

RANDOM ACCESS UNITS

MAGNETIC MEMORIES

WRITING AND READING

CORE-TO-CORE TRANSFER

FERRITE APERTURES

1. What is the relationship between the cost and access time of storage devices?
2. In terms of capacity, access time, and cost per bit of storage, how do auxiliary storage units compare with main memory units?
3. Why are registers said to have zero access time?
4. Data transfer between auxiliary storage units and the main memory unit requires the use of _____.
5. Which of the following storage devices is volatile? (a) A magnetic tape. (b) A punched card. (c) A delay line.
6. In a memory unit which uses destructive read-out, all the storage devices are destroyed by the read-out process. True or False?
7. The access time of a cyclic memory unit is made up of what two time intervals?
8. In the word-organized memory, each row of the matrix stores one word. True or False?
9. In the bit-organized memory, at least _____ selection signals are required to access each word.
10. List the types of magnetic storage devices found in large memory units.
11. Ferrite cores can be made smaller than alloy cores and are therefore capable of achieving _____ switching speeds.

12. A core made of 3-80 permalloy is composed of what percentages of molybdenum, nickel, and iron?
13. The 0 and 1 states in a magnetic core are represented by two different levels of magnetic flux circulating in the same direction. True or False?
14. In a magnetic core, the storage state is represented by the (a) saturation. (b) remanent flux density level.
15. In order to change the state of a magnetic core in a matrix arrangement, _____ pulses of magnitude $\frac{1}{2}I_m$ are required.
16. How are 1's and 0's read out of a magnetic core memory?
17. A basic magnetic memory clock cycle is composed of two phased clock pulses, one for (a) _____ and the other for (b) _____.
18. The read winding has less impedance when the core switches from the 1 to the 0 state than it does when the core does not switch. True or False?
19. What is the function of D_1 in the circuit of Figure 12A?
20. If core A of Figure 13A is in the 0 state and current I_r is present, the current through D_1 is almost equal to the current through D_2 . True or False?
21. Whenever core A of Figure 13A is switched from the 1 state to the 0 state, the 1 is transferred to core B. True or False?
22. When I_r is present, core B of Figure 14A is set to the 1 state if (a) core A is in the 1 state and core C is in the 0 state. (b) core A is in the 0 state and core C is in the 1 state. (c) both core A and core B are in the 1 state. (d) both core A and core C are in the 0 state.

Basically, magnetism results from a movement of electrons. This includes the movement of an electron in its atomic orbit and the "spin" which an electron has about a self-contained axis. Such a magnetic field is very small. In ferromagnetic materials, large groups of neighboring atoms tend to act together in a "cluster," called a **FERROMAGNETIC DOMAIN**. Many such groups are independently scattered throughout the material. They have a random orientation initially, but will "line-up" under the influence of a magnetizing force. The greater the force, the greater the "line-up" and strength of the resulting magnetic field. This is shown in Figure 15. If the major domain orientation in Figure 15A represents a logical 1, the domains must be oriented in the opposite direction to represent a logical 0. This may occur as the result of **DOMAIN WALL MOTION** (domain boundaries are called walls), where the domains already in the desired direction "grow" large enough to reverse the magnetic polarity (Figure 15B). The reorientation may also occur by **DOMAIN ROTATION**, where the magnetic fields or entire domains rotate opposite to the direction they were in when they represented a logical 1. This is shown in Figure 15C.

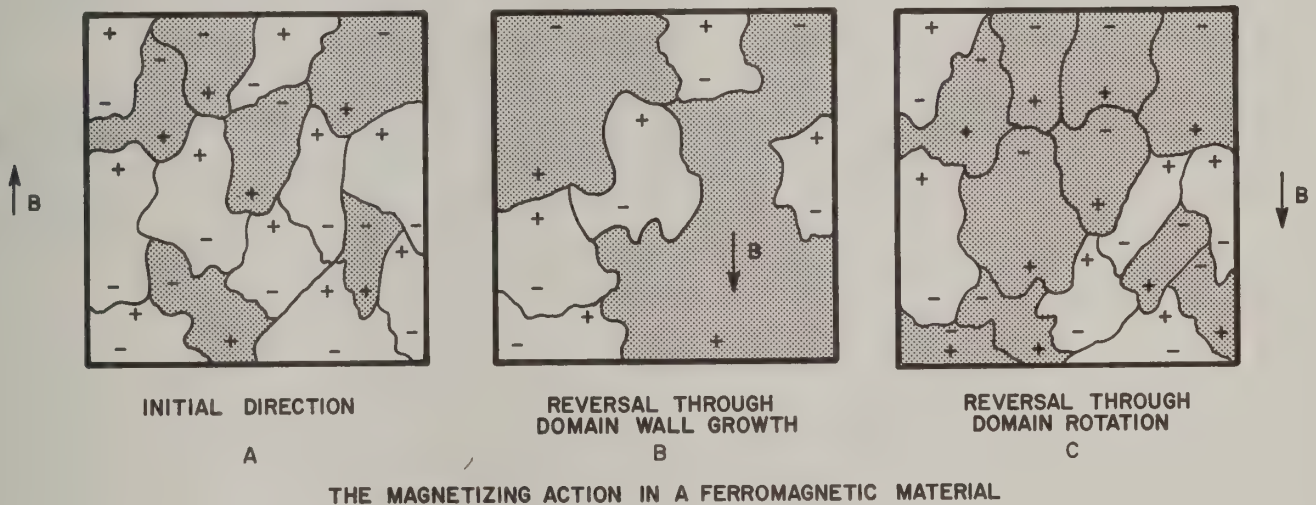


Figure 15

Switching takes place in a ferromagnetic material through a combination of domain wall motion and rotation. However, domain wall motion normally predominates. This accounts for the slow switching of magnetic cores. For thin-film memories, the magnetizing current enables domain rotation to predominate. This motion is limited only by the speed at which the individual electrons can respond to the magnetizing force.

The layout of a single thin-film memory cell is shown in Figure 16. The small "patch" of ferromagnetic film is deposited on the substrate in the presence of a constant magnetic field. This gives the film an initial magnetic field direction. Conductive layers of thin-film metallization are then deposited. The conductive layers consist of a drive conductor laid at right angles to the initial magnetic field direction, two information conductors,

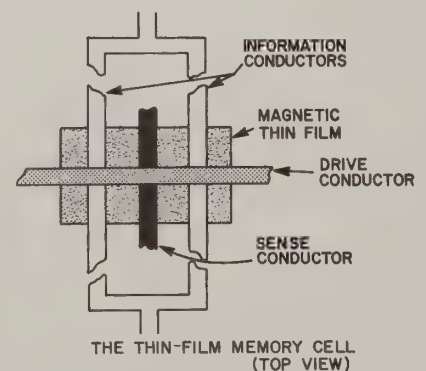
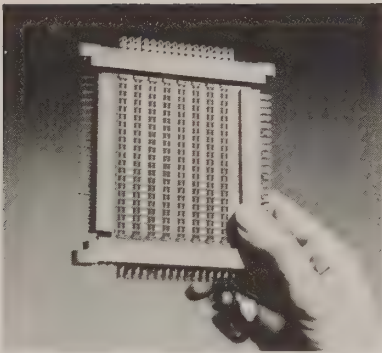


Figure 16



Typical Thin-Film Memory Plane
Courtesy Burroughs Corp.

Figure
17

and a sense conductor laid along the initial field direction. Domain rotation is assured by activating the drive conductor an instant before activating the information conductors. This rotates the domain 90° and "sets it up" for the complete rotation when the information signal is applied. The direction of the information current determines whether the domains will finally rest with a positive polarity toward the top or bottom of the patch, to establish a 1 or a 0. Each time a drive pulse switches the domains 90° , a voltage is induced in the sense conductor which reads out the last state of the cell. This voltage is either positive or negative, indicating a 1 or a 0, respectively. The read-out operation is destructive.

Figure 17 is a completed thin-film memory plane which can store 20 words of 8 bits each (a total of 160 bits). The 160 thin-film patches are visible (gray areas). The drive conductors can be seen running horizontally and the two information conductors can be seen as black strips running vertically on each side of the patches. The thin black strip running vertically between each pair (the sense conductor) is barely visible under close examination.

Superconductive or **CRYOGENIC THIN-FILM MEMORY** units are also available. These units are based on the fact that certain metals become superconductors at temperatures near absolute zero (-273°C). These metals include niobium, lead, tantalum, and tin. Cryogenic thin-film memory planes are immersed in liquid helium at -269°C . This complicates the design, use, and maintenance of such units in order to accommodate such a "container."

SUPERCONDUCTORS have very unusual characteristics. First, when a current is started in a superconductive loop, it will continue circulating indefinitely (when the temperature is near absolute zero). There is no need for any further application of a potential difference and there is no heat dissipated from the material. Second, if an external magnetic field is applied to a superconductor, the material reverts to its normal resistive characteristic. There is a "critical value" of magnetic field strength required to accomplish this reversion. In fact, a current circulating within a superconductive loop may, itself, destroy the superconducting characteristic if the self-induced field strength exceeds the critical value (the temperature still about absolute zero). Third, a superconducting loop can provide a perfect magnetic shield—magnetic flux lines cannot penetrate from one face of the loop to the other. All of these phenomena are used in creating the cryogenic thin-film memory cell.

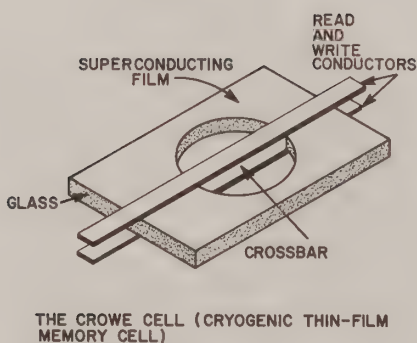
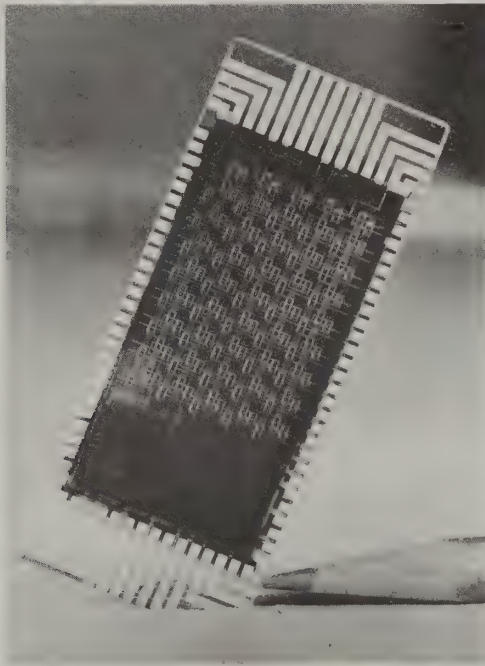


Figure
18

Figure 18 shows the basic construction of the **CROWE CELL**. This device is formed by depositing a thin film of superconductive material on a glass substrate. A small circular aperture is left "open." The small circular aperture is bisected by a narrow thin-film "crossbar." The edges of the open loop ultimately serve as a closed superconductive loop. Metallization, along the crossbar on opposite sides of the glass plate, forms the two conductors (one for write and one for read) on the cell.

To write a 1 into a Crowe cell, a two-level pulse is applied to the write-in conductor. The first level of the pulse initiates a current circulation around the edge of the superconducting aperture with shielding present for the read-out conduction. This current is large enough so, once circulating, its self-induced magnetic field exceeds the necessary critical value, thus causing the material to revert to its normal resistive state with no shielding present. It is during this short time that the read-out conductor can "sense" the last state of the cell. The second level of the pulse consists of reducing the write pulse in magnitude so the material returns to the superconducting condition. The cell again shields the read-out conductor from further variations. When the write pulse drops to zero, the circulating current continues in the loop, representing a logical 1. A logical 0 is written in a similar manner; however, this time with pulses in the opposite polarity. The read-out of the last state occurs as a new state is written in; therefore, the read-out is destructive.



*Typical Cryogenic Thin-Film Memory Plane
Courtesy IBM Corp.*

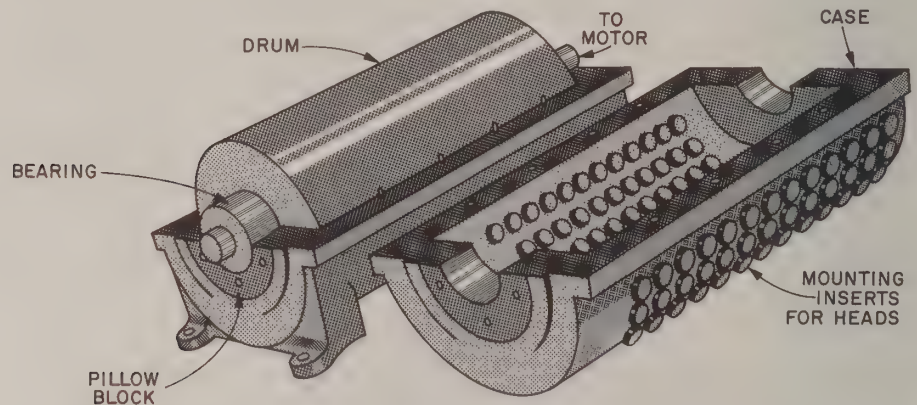
Figure 19

Figure 19 shows a cryogenic thin-film memory plane, about the size of a postage stamp. This particular cryogenic thin-film memory plane can store 40 bits. A large-capacity memory unit is formed by interconnecting the desired number of thin-film memory planes.

Magnetic Drums and Discs

A magnetic drum is basically constructed of a precision built cylinder coated with a magnetic material similar to the magnetic material used on

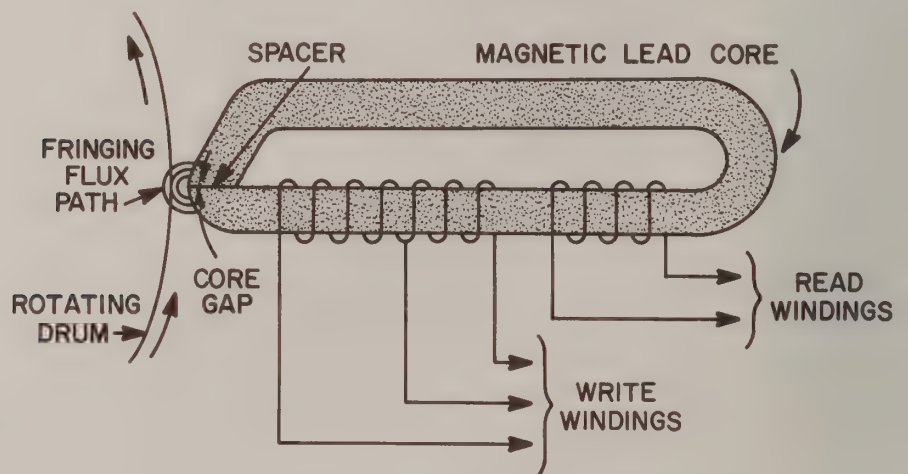
magnetic recording tapes. Typical magnetic drums range from 3 inches to 30 inches in diameter and from under 3 inches to 48 inches in length. The cast drum is locked to a steel shaft and rotates on its axis. Large drums use aluminum shafts with press fitted steel end caps for bearing seats. Magnetic recording heads are placed along the length of the drum so that each head covers a narrow band or track around the circumference of the drum. Mounting inserts for the heads are pressed into a cast aluminum case or shroud.



BASIC COMPONENTS OF MAGNETIC DRUM MEMORY

Figure 20

Figure 20 shows the case and drum assembly. The drum is totally enclosed in the case, with the rotor bearings supported in pillow blocks machined in the case. The drum is driven by a motor connected to one end of its shaft.



THE MAGNETIC DRUM READ-WRITE HEAD

Figure 21



*Large Magnetic Drum Memory Unit
Courtesy Philco Corp., Computer Div.*

Figure 22

Figure 21 shows the basic structure of a magnetic recording head. The head consists of a core structure linked with both reading and writing coils. The write winding is center tapped so that the direction of current through the winding can be reversed by selection of the end taps. Current in one direction writes a 1 on the surface of the drum; current in the other direction writes a 0 on the surface of the drum. The head is mounted in the case surrounding the drum and is adjusted to bring the core gap close to the drum. During the writing operation, fringing flux around the core gap magnetizes a spot on the drum surface. During read-out, voltages are induced in the read winding as the magnetized spots pass by the head.

Figures 22 and 23 illustrate magnetic memory drums for a digital computer. The motor appears at the top in Figure 22. Figure 23 shows the large number

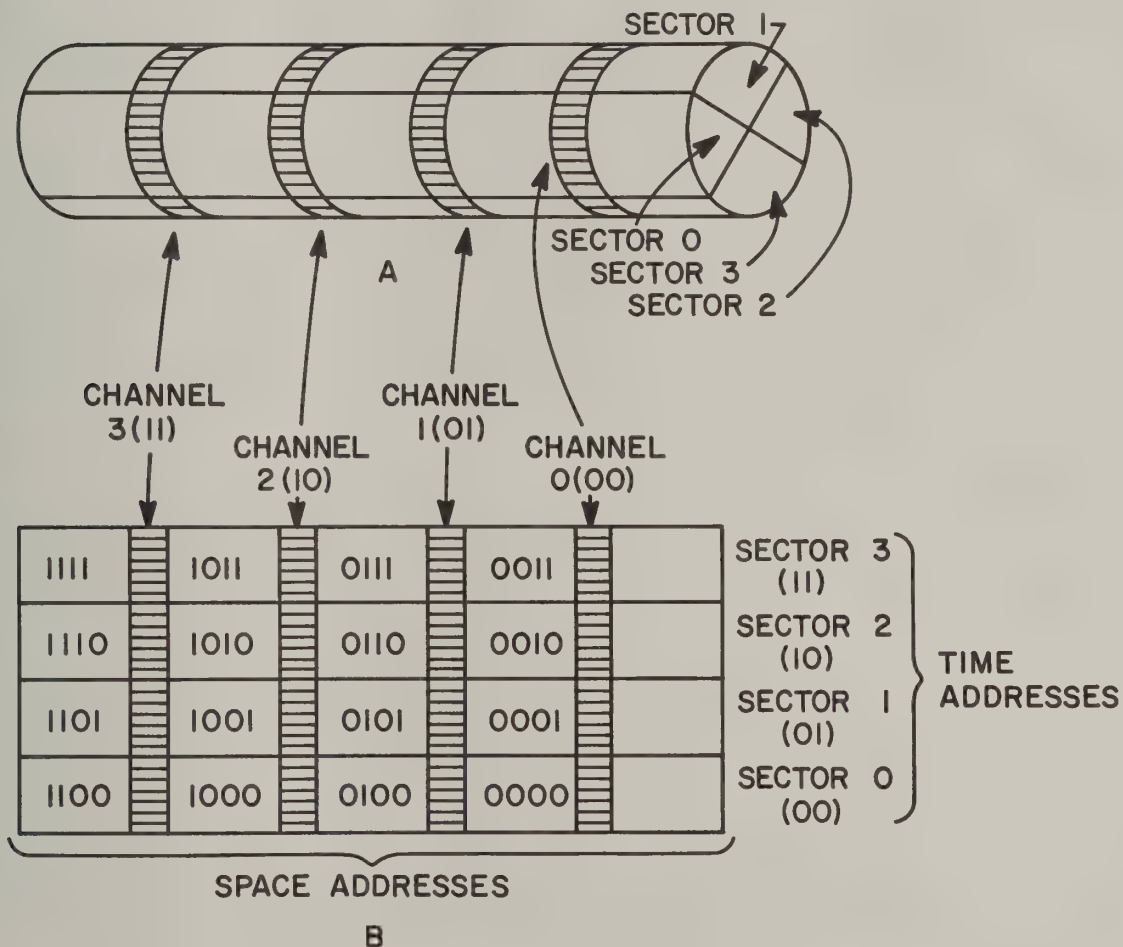


*Large Magnetic Drum Memory Unit
Courtesy Bryant Computer Products*

Figure 23

of read-write heads placed around the circumference of a large memory drum.

Data may be stored on a magnetic drum in either serial or parallel form. In serial form, all of the bits in a word are recorded one after the other on a single track. In parallel form, the bits in a word are recorded on separate, neighboring tracks.



STORAGE ARRANGEMENT AND ADDRESSING
FOR A MAGNETIC DRUM MEMORY

Figure 24

Figure 24A shows the arrangement of a 16-word serial memory unit. Figure 24B shows where the words are located and how they are addressed. This is accomplished by laying out the surface of the drum on a flat surface. The drum contains four tracks and is divided into four **SECTORS**. Each sector contains one full computer word. For this example four words can be recorded on each track—one in each sector. Each word is made up of

verts the output of BR to a form suitable for recording on the drum surface. The write amplifier is a gated amplifier which requires bit markers from AND gate 3 for proper operation.

At the end of the writing operation, the register contents will have been recorded on the drum. FF_2 is reset by the next word marker from amplifier 1. When FF_2 is reset, AND gate 2 is disabled and FF_1 is also reset. When FF_1 is reset, it produces an operation-complete signal which is fed through the control unit to the other computer circuits.

The read operation is similar to the write operation except that the control unit feeds read signals to AND gates 4 and 6. With AND gates 4 and 6 enabled, AND gate 7 is also enabled, and the word being read passes through amplifier 3 and AND gate 7 and is shifted into BR. The existing contents of BR are shifted out of the right end. They cannot be recorded on the drum since no bit markers were fed to the write amplifier. The read operation is ended in the same manner as the write operation.

To insure that the current-address counter C is correctly set, one of the word markers is split in two, providing two pulses very close together at the output of amplifier 1. If the counter has the wrong relationship to the word markers, the extra pulse advances the counter one extra count for each revolution of the drum. When the counter is correctly set, in relation to the word markers, a blocking pulse (produced by the last counter stage when it is reset) is fed to amplifier 1 from the counter. This blocking pulse momentarily blocks amplifier 1, not allowing the extra pulse to pass through.

Discs, which look like phonograph records, are also covered with a magnetic coating and used in memory units. They operate similar to magnetic drums except for a few mechanical differences due to their shape. A stack of discs can be mounted together in a cabinet with one or more magnetic heads. Such a cabinet looks similar to a coin-operated, automatic record player (juke box). In some units, all the discs turn continuously. A head moves along a shaft until it comes to the desired disc, then moves inward to the desired channel. In other units, each disc only revolves when it is selected. The discs may also be arranged with their centers located on a circle, so that a head swinging on a pivoted arm can read each disc.

INTEGRATED CIRCUIT MEMORIES

Recently there has been considerable development of the IC chip in terms of memory cells. An obvious way to achieve a random access IC memory is simply to fabricate the necessary number of flip-flops on the chip, then interconnect these in a matrix fashion with X-Y metallization paths. This

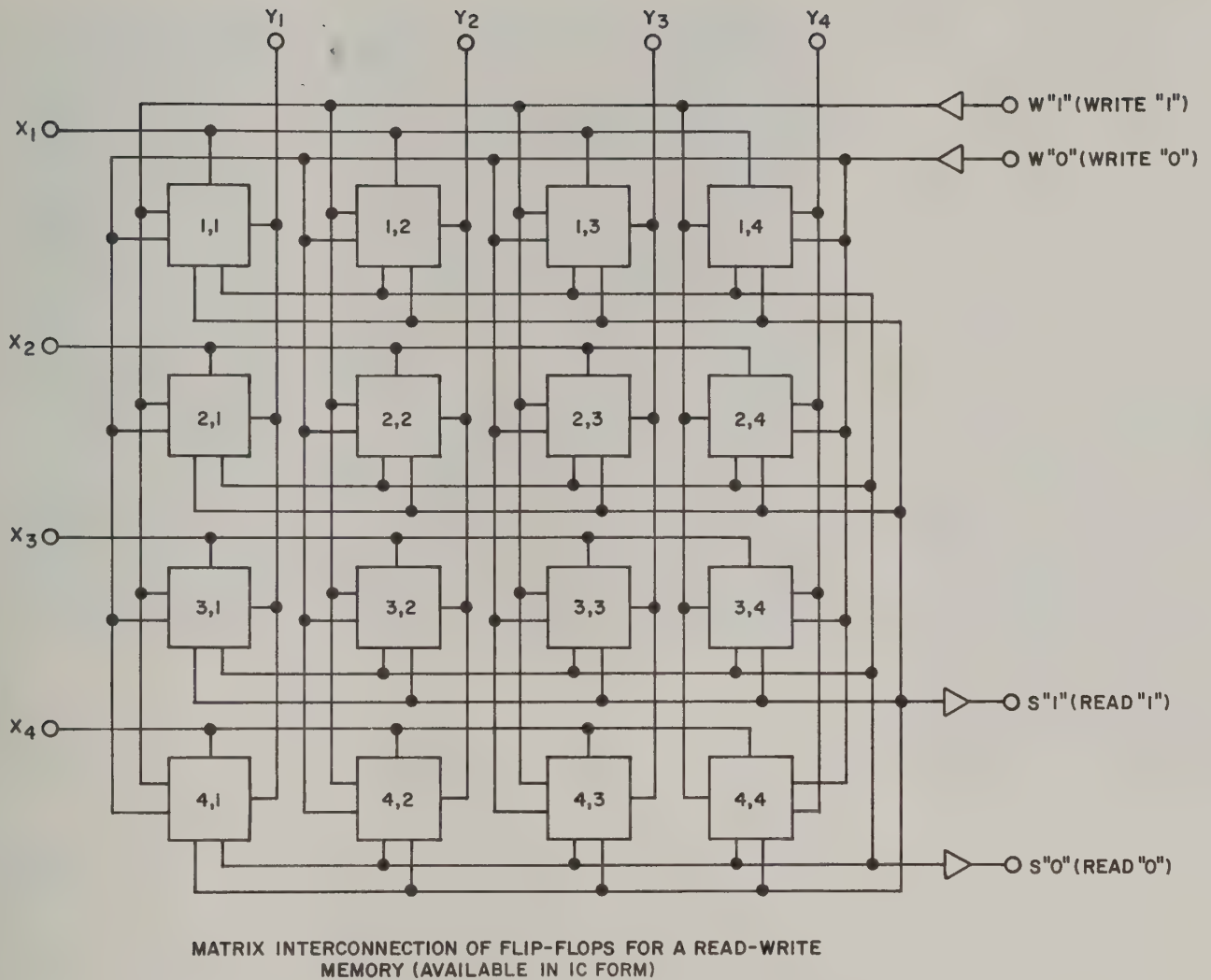
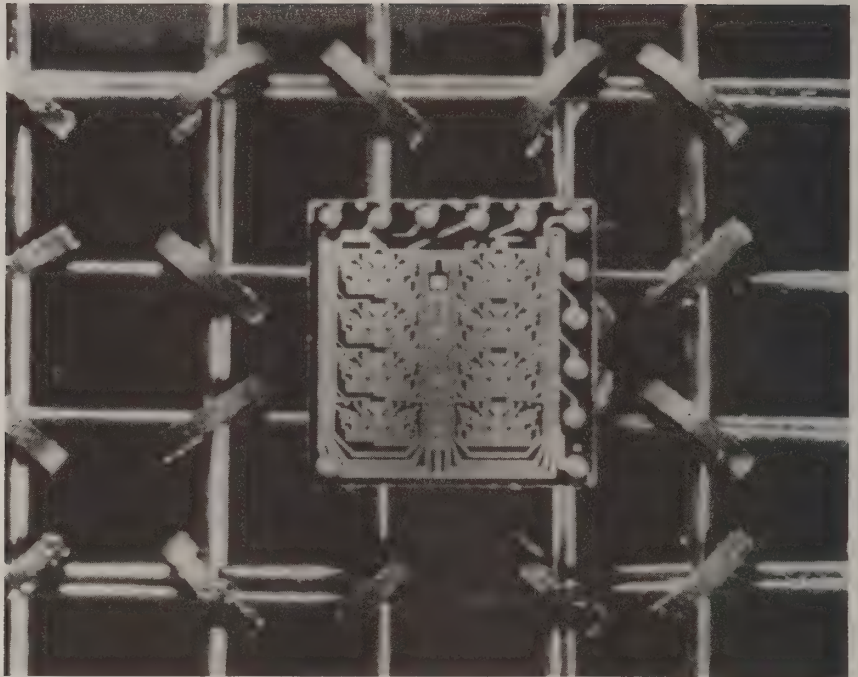


Figure 26

approach is being taken (as shown in Figure 26). With NPN transistors used as active devices, memory chips of 4, 8 and 16 bit capacity are available. Figure 27 compares the size of a 16-bit memory IC with a conventional ferrite core memory plane. The IC chip occupies the area equivalent to a 4-bit capacity in the more conventional memory. The use of metal-oxide-semiconductor field effect transistors (MOSFET's) allows an even greater capacity. Random access memory MOS IC's are available with capacities from 32 to more than 2000 bits. Although the term RAM can be used to refer to random access memories, it is most commonly used to indicate an IC random access memory unit.

MOS RAM's are ideally suited for those applications requiring a small- to medium-size storage capability, high speeds, and low power dissipation.



*Comparison of IC and Ferrite Core Memory Sizes
Courtesy IBM Corp.*

Figure 27

They can replace core memories for small computers, offering much faster access times and simpler drive circuitry. MOS RAM's may be either static or dynamic. In the static types, MOSFET flip-flops are used to store the information. The read-out is non-destructive, thus producing a memory which is lost only if the power is turned off. In the dynamic type, clock signals are used to continuously circulate data. If these signals are lost, the memory is lost.

RAM's using conventional transistor structures have access times on the order of a microsecond. MOS RAM's can be accessed in as little as 200 nanoseconds (200×10^{-9} sec.).

Many memory units in special purpose computers are READ-ONLY MEMORIES (ROM). ROM units "store" a specific set of instructions or operations. These instructions or operations are then read out for control purposes, when required. The information stored in a ROM is permanently "programmed" into the memory at the time of manufacture. Once the information is entered, it cannot be changed. However, it can be read out as often as desired. The read-out is non-destructive and the stored data non-volatile.

Common applications for ROM's include the table look-up of mathematical values, conversion from one binary code to another, or the conversion of a binary code into alpha numeric characters displayed with nixie tubes or matrix displays. For example, whenever a square root is needed (Figure 28) the value may be fed into a ROM. A range of square roots is permanently stored in the ROM and the output will always be the square root of the input, as long as it is within this range.



USE OF A ROM PROGRAMMED FOR THE SQUARE ROOT OPERATION

Figure
28

In the ROM, a single MOSFET is used as a memory cell. During manufacturing, any desired MOSFET can be made to permanently represent a 0 by making it inoperative. Therefore, when the unit is energized, only the operable transistor positions represent logical 1's, all other positions are logical 0's. The MOSFET's are arranged in an X-Y matrix similar to Figure 29 and can be addressed in a random fashion.

MOS ROM's are available with capacities ranging from 256 to over 4000 bits. Access times as small as 50 nanoseconds are typical.

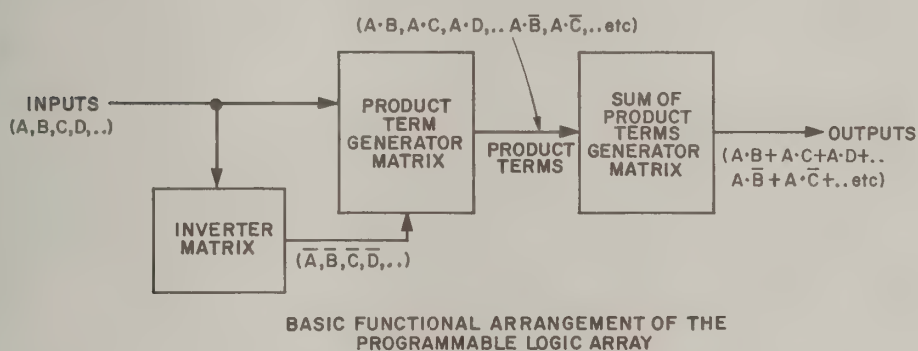
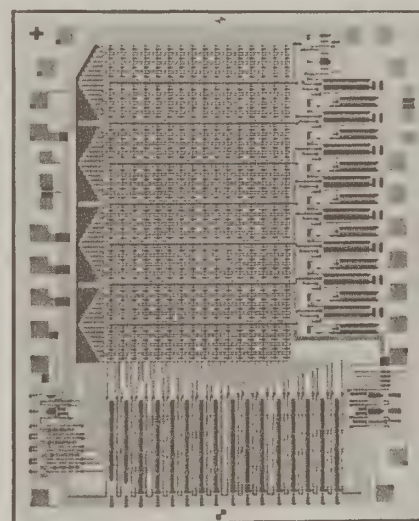


Figure 30

One of the more recent developments in IC memories is the **PROGRAMMABLE LOGIC ARRAY (PLA)**. The PLA structure is extremely simple. Any logic function can be written as a sum of products or as a product of sums. This is exactly what is implemented on the PLA chip. A desired number of leads feed the chip with inputs (A, B, C, D . . .), as shown in Figure 30. The chip contains an inverter for each input; therefore, all inputs and their complements ($\bar{A}$, $\bar{B}$, $\bar{C}$, $\bar{D}$. . .) are available. The **PRODUCT TERM GENERATOR MATRIX** is a set of AND gates which generates all of the product terms: $A \cdot B$, $A \cdot C$, $A \cdot D$. . . $A \cdot \bar{B}$, $A \cdot \bar{C}$, . . . etc. The **SUM OF PRODUCT TERMS GENERATOR MATRIX** is a set of OR gates which combines all of the product terms into sum terms ($A \cdot B + A \cdot C + A \cdot D + \dots A \cdot \bar{B} + A \cdot \bar{C} + \dots$). With this kind of arrangement, any desired logic function can be implemented.



A 1024-Bit Read Only Memory Chip
Courtesy Ziff-Davis Publishing Co.

Figure
29

An alternative PLA arrangement would generate the sum terms ($A + B$, $A + C$, $A + D \dots A + \bar{B}$, $A + \bar{C}$, $\dots$ etc.) first, then the product terms $[(A + B)(A + C)(A + D) \dots (A + B)(A + C) \dots]$.

PLA's may also include sets of J-K flip-flops to provide various forms of storage capability along with the logic functions.

This concept is extremely versatile, and may well be the key to a revolution in solid state computer systems.

SUMMARY

Although many different physical properties have been used to store data, magnetism is still the most common physical property in use for storing large amounts of data. Magnetic hysteretic devices such as toroidal cores, ferrite apertured plates, and magnetic thin films are used in word-organized and bit-organized random access memories. Other types of magnetic memories are tapes, drums, and discs.

In random access memory units, the data are fixed in space with respect to the read-write stations. This is called storage in space. The process of selecting a particular address in this type of memory unit is called S-selection. Random access memories are noncyclic. In cyclic storage systems, the data are in continuous motion with respect to the read-write stations. This is called storage in time. The process of selecting a particular address in this type of memory unit is called T-selection.

There are two types of random access memory units: word-organized and bit-organized. In word-organized memories, the memory cells are arranged in rows and columns. Each row stores one word. To select a particular row, a signal is applied to the desired row and the word stored at that address appears on the column lines. To write in a word, the word to be written is fed to the column lines when the selecting signal is applied to the desired row.

In bit-organized memories, a number of rectangular arrays (planes) are stacked one behind the other. One bit of a word is stored in each plane. Coincident current methods may be used to read out this type of memory. Address selection is accomplished by applying two signals, one each to the desired horizontal (X) and vertical (Y) lines in each plane. Only the memory cells at the intersections of the X and Y lines are selected.

The following Practice Exercise questions cover the subjects which you have just studied. They are:

MAGNETIC MEMORIES

THIN-FILM MEMORIES

MAGNETIC DRUMS AND DISCS

INTEGRATED CIRCUIT MEMORIES

23. Compare the cycle time for thin-film and magnetic core memories.
24. Describe the two ways in which a ferromagnetic material may be caused to assume a directional magnetic field.
25. Describe how the logical state of a thin-film memory cell is changed.
26. How is the state of a thin-film memory cell read out?
27. What temperature values are required in order to achieve superconductivity?
28. Describe the three unusual characteristics of a superconductive loop.
29. Why is a two-level pulse used to cycle the Crowe cell?
30. What is a primary disadvantage of cryogenic thin-film memory units?
31. The head for a magnetic drum memory may contain either a read or a write coil. True or False?
32. Data may be stored on a magnetic drum in either serial or parallel form. True or False?
33. How many addresses are required in order to gain access to a word on a magnetic drum?
34. What is the purpose of the prerecorded clock markers on a magnetic drum?
35. When EC in Figure 25 produces a start pulse, (a) a word is always read out of the drum regardless of the state of FF_2 . (b) the counter resets to the 0 state. (c) the desired address, as indicated by the number stored in AR, is about to pass under the read-write head.
36. If a word in the buffer register BR of Figure 25 is to be written onto the drum, when is it shifted out of BR?

37. In order to read data from the drum, which AND gates in Figure 25 are enabled by the read signal?
38. Why is it impossible for data to be written onto the drum in Figure 25 when data is shifted out the read end of BR?
39. What advantages do MOS RAM's offer over core memories?
40. How do dynamic MOS RAM's differ from static MOS RAM's?
41. ROM's are used to implement small to medium scale operations that never change and are used over and over in computer systems. True or False?
42. How is it possible to implement any desired logic function with a programmable logic array?

The read-out signals appear on sense windings which connect all the memory cells in each plane. Write-in is accomplished by applying the correct signals to other windings which connect all the memory cells in each plane.

The two types of magnetic cores used in digital applications are: molded ferrite cores and tape-wound cores. These cores may be magnetized to one of two remanent states called the positive remanent state ($+B_r$) and the negative remanent state ($-B_r$). One remanent state represents 0 and the other remanent state represents 1.

Unlike the magnetic cores used in most electron circuits, the magnetic cores used in digital applications have almost square hysteresis loops. The top and bottom of a square hysteresis loop is almost horizontal. The remanent flux density of a square loop core remains almost constant for a range of driving current values. This produces saturation in either direction.

Core gates can be formed using the three basic transfer loops: the single-diode loop, the split-winding loop, and the inhibit loop. The single-diode loop provides an unconditional transfer of data (either 0 or 1) from one core to another. The split-winding loop provides a conditional transfer of data from one core to another. The inhibit loop is a special form of split-winding loop in which the output of one core inhibits the transfer of data between two other cores.

In magnetic core storage units, data are written into or read out of the addresses by coincident-current selection. Using this method, magnetizing forces great enough to switch a core from one state to the other are produced only when two currents in the same direction occur simultaneously (are coincident). In magnetic core storage units, the cores are arranged in rows and columns called a matrix. Packaged commercial matrices are called planes (or frames). Complete magnetic core storage units are formed by stacking these planes one behind the other. There is one plane for each bit in a word. Succeeding bits in a word are stored in the same core location but in neighboring planes.

Ferrite aperture planes may be used in the place of magnetic core planes. They are sheets of ferrite material with a matrix of perforations. Each hole in the sheet acts as a memory cell.

Thin-film memories use metal films deposited onto a glass substrate. They can be cycled in less time than a magnetic core. Cryogenic thin-film mem-

ories are cooled to near absolute zero. These memories require extremely small driving currents and occupy a very small volume.

Some memory units, such as magnetic drums, use a combination of S-selection and T-selection. In magnetic drum memory units, S-selection is used to select the desired track. T-selection is used to select the desired sector of the selected track.

There are two types of T-selection. In one type, the addresses of the word locations are recorded at the locations. These addresses are read out and compared with the desired address. When the two are equal (coincident), the desired address is about to pass the read-write station. In the other type of T-selection, the addresses increase in numerical order and a counter is used to indicate which address is about to pass the read-write station.

Integrated circuits are rapidly taking over many of the storage applications previously reserved for magnetic storage devices. This is especially true in areas requiring a small to medium size storage capacity. Random access memory IC's (RAM's) are available in both static and dynamic forms. Access times on the order of 200 nanoseconds are typical. Read-only memory IC's (ROM's) have information permanently "stored" on the chip at the time of manufacture. Once formed, the data cannot be changed; however, it can be read out as often as desired. IC's containing programmable logic arrays (PLA's) are capable of generating any desired set of logic functions by using matrices of AND and OR gates. Storage capability, using J-K flip-flops, can also be built into these chips.

IMPORTANT DEFINITIONS

ACCESS TIME—The time required to transfer data between the memory unit and other units in a computer.

ADDRESS—A location where information is stored.

AUXILIARY STORAGE UNITS—Secondary memory units within various parts of a computer.

BIT-ORGANIZED MEMORIES—Memories constructed so that each bit must be addressed individually.

COINCIDENT-CURRENT MEMORIES—BIT-ORGANIZED MEMORIES.

COINCIDENT MEMORIES—BIT-ORGANIZED MEMORIES.

CONDITIONAL TRANSFER—A transfer of information which can take place only when a specific set of conditions is satisfied.

CROWE CELL—A superconductive memory cell containing a circular aperture crossed by a thin strip of superconductive material called a crossbar. The read and write conductors are thin strips in parallel with the crossbar, one on each side of the aperture. The Crowe Cell is also known as a trapped flux memory cell.

CRYOGENIC THIN-FILM MEMORY—A memory unit composed of magnetic thin films (such as nickel-iron alloy) on a glass substrate and operated in a superconductive state.

CYCLIC STORAGE—A memory unit which must sequence through all lower-numbered address locations in order to reach a specific address.

DESTRUCTIVE READ-OUT—The loss of existing stored data when the memory states are read out.

DOMAIN ROTATION—A process where the magnetic field of a material changes direction through the rotation of its FERROMAGNETIC DOMAINS.

DOMAIN WALL MOTION—A process where the magnetic field of a material changes direction through the “growth” of existing FERROMAGNETIC DOMAINS.

DYNAMIC STORAGE—A memory arrangement where data continuously change position.

IMPORTANT DEFINITIONS (Continued)

EYEBROW SYMBOL—A curved line within a memory logic symbol indicating the conditions which must exist before data transfer can occur.

FERRITE APERTURED PLATES—Sheets of ferrite material which have been perforated to form a matrix of memory cells. Each hole serves the same function as a magnetic core.

FERRITE CORE MEMORY—A unit utilizing small ferrite rings with circulating fluxes to represent logical states.

FERROMAGNETIC DOMAIN—The atomic cluster within a material responsible for the magnetic properties of the material.

INHIBIT TRANSFER LOOP—A special form of the SPLIT-WINDING TRANSFER LOOP which inhibits the transfer of information under a given condition.

LATENCY TIME—The time required for a CYCLIC STORAGE unit to sequence through the lower-numbered addresses to arrive at the desired address.

MAIN MEMORY UNIT—The primary internal data storage unit in a computer.

MATRIX—An arrangement of components in a row and column fashion. Storage matrices are also called frames or planes.

MEMORY CELL—A device capable of being placed in a logical 1 or logical 0 state, and capable of retaining this state until changed electrically.

NON-DESTRUCTIVE READ-OUT—The read-out of data in a memory unit without changing or destroying the stored data.

NON-VOLATILE STORAGE—Permanent storage capability; the data is not lost over an indefinite time period.

PERMANENT STORAGE—See NON-VOLATILE STORAGE.

PRODUCT TERM GENERATOR MATRIX—A group of AND gates that produces all of the product terms.

PROGRAMMABLE LOGIC ARRAY (PLA)—An IC unit which is capable of generating any desired logic function on the basis of a sum of products, or a product of sums.

IMPORTANT DEFINITIONS (Continued)

RAM—Commonly applied to an IC random access memory unit. Random access memory is the same as random access storage.

RANDOM ACCESS STORAGE—See RAM.

READ ONLY MEMORIES (ROM)—Commonly applied to an IC that has information permanently written into it and is therefore used only for read-out.

REMANENT FLUX DENSITY—The magnetic flux density remaining in a material after it has reached saturation after the magnetizing force is removed.

SECTORS—Geometric divisions of a magnetic drum, each containing one word.

SENSE WIRES—Separate conductors, used in some matrix memory arrangements, permitting the read-out of a MEMORY CELL state.

SINGLE-DIODE TRANSFER LOOP—A circuit arrangement permitting UNCONDITIONAL TRANSFER of information.

SPLIT-WINDING TRANSFER LOOP—A circuit arrangement permitting a CONDITIONAL TRANSFER of information.

S-SELECTION—The static selection of an address in a fixed spatial location.

STATIC STORAGE—A memory arrangement where data are in fixed locations.

SUM OF PRODUCT TERMS GENERATOR MATRIX—One or more OR gates that combine all of the terms from the PRODUCT TERM GENERATOR MATRIX.

SUPERCONDUCTORS—Certain materials which display a “zero resistance” characteristic, along with other “unusual” properties, at temperatures near absolute zero.

THIN-FILM MEMORY—A memory unit composed of magnetic thin films of a thin glass substrate.

TRANSFER LOOPS—Circuits which allow information transfer between storage units.

T-SELECTION—The dynamic selection of an address in a moving spatial (time) location.

IMPORTANT DEFINITIONS (Continued)

UNCONDITIONAL TRANSFER—A transfer of information which may take place under any conditions.

VOLATILE STORAGE—Storage units which lose their data when the power is removed.

WORD-ORGANIZED MEMORIES—Memories constructed so that one address locates a complete data word.

PRACTICE EXERCISE SOLUTIONS

1. Faster storage devices are usually more expensive than slow storage devices.
2. In terms of total capacity, auxiliary storage units are larger than the main memory. Auxiliary units are also slower and have a lower cost per bit.
3. The access time of most registers is negligible compared to main and auxiliary unit access times; therefore, they are said to have zero access time.
4. buffers
5. (c) a delay line—The removal of power does not destroy the data on a magnetic tape or on a punched card.
6. False—The stored data is destroyed, not the storage devices.
7. Cyclic memory access time is made up of the time it takes to reach the desired address (latency time) and the time required to read the word at that address (one word time).
8. True
9. two
10. Magnetic cores, ferrite apertures, drums, discs, and thin-film memories all utilize magnetization for their basic operation.
11. higher
12. 3-80 permalloy is composed of 3% molybdenum, 80% nickel and the rest (17%) is iron.
13. False—The flux circulates in opposite directions.
14. (b) remanent flux density level.
15. two
16. Read-out is accomplished with a set of negative X- and Y-pulses. If a core is in the 1 state, it switches to the 0 state and produces a large current pulse in the sense wire. If a core is in the 0 state, it produces an insignificant sense wire pulse which is ignored by the system.
17. (a) write-in (b) read-out

PRACTICE EXERCISE SOLUTIONS (Continued)

18. False—The read winding has more impedance when the core switches from the 1 to the 0 state than it does when the core does not switch.
19. D_1 blocks current in the direction opposite to I_t and opposes current produced by voltages induced in the write winding of core B.
20. True
21. False—The 1 is transferred to core B only if I_t is responsible for switching core A to the 0 state. If current I_o switches core A to the 0 state, the 1 is not transferred to core B. This produces a conditional transfer.
22. (a) core A is in the 1 state and core C is in the 0 state.
23. Thin-film memory cycle times are less than one microsecond, while a few microseconds are required for magnetic core cycling.
24. Ferromagnetic material assumes a directional magnetic field through domain wall motion; where the domains already in the desired direction “grow” large enough to reverse the magnetic polarity; and through domain rotation, where the magnetic fields of entire domains rotate to the opposite direction.
25. The change is accomplished through domain rotation. This is assured by activating the drive conductor an instant before activating the information conductor. Since the two are at right angles, this rotates the domains 90° , then completes the rotation when the information signal arrives.
26. Each time a drive pulse switches the domains 90° , a voltage is induced in the sense conductor to read out the last state of the cell.
27. Temperatures near absolute zero (-273° C) are required in order to achieve superconductivity.
28. Once a current is started in a superconductive loop, it will continue circulating indefinitely. External magnetic fields above a critical value cause the material to revert to its normal resistive characteristic. A superconducting loop acts as a perfect magnetic shield.
29. The first level initiates a current flow in the superconductive loop. This flow is large enough to create a magnetic field of sufficient strength to revert the cell to its resistive state. This allows read-out of the last cell state to occur. The second level drops the magnetic field below the critical value, returning the loop to its superconductive condition and leaving the necessary circulating current in indefinite storage.
30. Cryogenic thin-film memory units require complex cooling equipment.

PRACTICE EXERCISE SOLUTIONS (Continued)

31. False—The head for a magnetic drum memory contains both a read and a write coil.
32. True
33. Two addresses—a T-address and an S-address—are needed to select a word from a magnetic drum.
34. The prerecorded clock markers on a magnetic drum serve two purposes: to indicate the bits and words.
35. (c) the desired address, as indicated by the number stored in AR, is about to pass under the read-write head.
36. The word to be written on the drum is shifted out of BR immediately after EC produces a start pulse.
37. AND gates 4 and 6 in Figure 25 are enabled in order to perform the read operation.
38. Data cannot be written onto the drum in Figure 25 when it is being shifted out of the read end of BR, since there are no bit markers being applied to the write amplifier from AND gate 3.
39. MOS RAM's offer faster access times and simpler drive circuitry than do core memories. Also, MOS RAM's provide a non-destructive read-out.
40. Static MOS RAM's store information in MOSFET flip-flops, while dynamic MOS RAM's provide storage by circulating the data.
41. True
42. Any desired logic function can be represented as a sum of products or product of sums of the inputs and their complements. Programmable logic arrays offer inverters, AND gate matrices and OR gate matrices to generate such terms. Furthermore, J-K flip-flops can be provided on the chip in order to implement desired storage capabilities.



ONE OF THE

BELL & HOWELL SCHOOLS

EXAMINATION
CHECK SHEET

1. D - COMPARING MAIN MEMORY UNITS WITH STORAGE REGISTERS, MAIN MEMORY UNITS HAVE -- LONGER ACCESS TIME, BUT LOWER COST PER BIT.

The type of storage device used to implement a memory unit of the required capacity is chosen through a compromise between two factors: Access Time and Cost. Access time is the time required to transfer data between the memory unit and the other units in the computer. Fast storage devices are usually more expensive than slow storage devices. The main memory unit is made up of less expensive slow storage devices; therefore, it has a longer access time and a lower cost per bit than the faster storage devices.

2. C - IN RANDOM ACCESS MEMORY UNITS, THE ACCESS TIME -- DOES NOT DEPEND ON THE PREVIOUS ADDRESS SELECTED.

The access time of a cyclic memory unit is made up of the latency time plus the word time. Cyclic memory also depends on the previous address selected. No access time is negligible. Only in random access memory units is the access time not dependent on the previous address selected.

3. D - TO SELECT A PARTICULAR ADDRESS IN A COINCIDENT-CURRENT MEMORY UNIT, AT LEAST -- TWO SELECTION SIGNALS ARE REQUIRED.

To find any address in a coincident-current memory unit, at least two selection signals are required: one signal for the X selection line, one for the Y selection line. At the intersection of the X and Y selection lines lies the requested address.

4. D - TO STORE DATA IN A MAGNETIC CORE, THE TWO BINARY STATES (0 AND 1) ARE REPRESENTED BY THE -- REMANANT FLUX DENSITIES OF THE CORE.

Once a positive current pulse of sufficient amplitude to saturate the core is applied, a remanant flux density continues circulating even after the pulse is removed. This level of flux density represents the 1 state. A similar condition occurs if a negative current pulse causes core saturation. However, this time the remaining flux density level represents circulation in the opposite direction.

5. B - IF THE WORDS IN A DIGITAL SYSTEM ARE MADE UP OF 35 BITS, THEN A MAGNETIC CORE STORAGE UNIT BUILT TO STORE THESE WORDS MUST HAVE -- 35 PLANES.

In bit-organized memories a number of rectangular planes are stacked one behind the other. One bit of a word is stored in each plane; therefore, a 35 bit word requires 35 planes.

6. A - WHEN A MAGNETIC CORE IS USED AS A CONTROLLABLE TRANSFORMER, AT LEAST -- 3 WINDINGS ARE REQUIRED.

A magnetic core may be used either as a controllable transformer or a variable impedance. When a core is used as a controllable transformer, at least three windings are required: a write, a read, and a sense. Current in the write winding switches the core to the 1 state, inserting a bit of data. Current in the read winding clears (switches) the core to the 0 state inducing a voltage in the sense winding. The voltage induced in the sense winding before the application of the read current is the output which indicates whether the core was in the 0 or 1 state.

7. B - THE SPEED OF DOMAIN ROTATION IN A FERROMAGNETIC MATERIAL IS -- LIMITED ONLY BY THE SPEED WITH WHICH THE SMALL MAGNETIC FIELDS OF SINGLE ELECTRONS CAN MOVE.

Reorientation of domains may occur by Domain Rotation, where the magnetic fields or entire domains rotate to the opposite direction. Switching takes place in a ferromagnetic material through a combination of domain wall motion and rotation. Domain wall motion normally predominates. This accounts for the slow switching of magnetic cores. For thin-film memories, the magnetizing current enables domain rotation to predominate. This motion is limited only by the speed at which the individual electrons can respond to the magnetizing force.

8. D - A CROWE CELL PROVIDES MAGNETIC SHIELDING BETWEEN THE READ AND WRITE CONDUCTORS WHEN -- IT IS IN ITS SUPERCONDUCTIVE STATE.

To write a 1 into a Crowe cell, a two-level pulse is applied to the write-in conductor. The first level of the pulse initiates a current circulating around the edge of the superconducting aperture. This current is large enough so, once circulating, its self-induced magnetic field exceeds the necessary critical value, thus causing the material of the Crowe cell to revert to its normal resistive state. The second level of the pulse consists of reducing the write pulse in magnitude so the material returns to the superconducting condition. The cell now shields the read-out conductor from further variation.

9. 'B - IN THE MAGNETIC DRUM MEMORY UNIT OF FIGURE 25, A WORD IS SHIFTED OUT OF BR AND RECORDED ON THE DRUM IMMEDIATELY AFTER -- EC PRODUCES A START PULSE. The control unit feeds a control pulse to FF₁. This sets FF₁ and enables AND gate 1 so the start pulse from EC can pass to FF₂. C is the clock and it does not produce a blocking pulse. AND gates 3, 5, and 6 are enabled by a write signal from the control unit. A word is shifted out of BR and recorded on the drum immediately after EC produces a start pulse. Thus, the bit markers from AND gate 2 pass through the buffer register shift circuits, causing the contents of BR to shift with each bit marker.

10. A - INTEGRATED CIRCUIT MEMORIES -- CAN BE OBTAINED WITH RANDOM-ACCESS READ-WRITE AND WITH READ-ONLY CAPABILITIES.

IC Memories are quite easily fabricated in the random-access form, and are quite popular because they provide many circuits in a relatively small surface area. Other memory units can be used compatibly with IC logic elements

QUESTIONS

IMPORTANT—These instructions **MUST** be accurately followed to avoid loss, or errors in grading.

Indicate your answer on this sheet by filling in the box for the most correct answer to each question, as illustrated in the example below.

When all questions have been answered, place the answer card in the proper position to line up the boxes on the card with the boxes on the sheet.

Next, copy the complete lesson code into the space provided on the card, and fill in the answer boxes to correspond with those previously filled in on this sheet.

Before mailing, be certain your correct student number, name and address appear on the card.

LESSON CODE
5250A

Example: A robin is
 A ☐ (A) a reptile. (B) an insect.
 B ☐
 C ☒ (C) a bird. (D) a mammal.
 D ☐

1. A ☐ Comparing main memory units with storage registers, main memory units have
 B ☐ (A) longer access time and higher cost per bit. (B) shorter access time and lower cost per bit. (C) shorter
 C ☐ access time, but higher cost per bit. (D) longer access time, but lower cost per bit.
 D ☒

2. A ☐ In random access memory units, the access time
 B ☐ (A) is equal to the latency time plus the word time. (B) depends on the previous address selected. (C)
 C ☒ does not depend on the previous address selected. (D) is longer than in cyclic memory units.
 D ☐

3. A ☐ To select a particular address in a coincident-current memory unit, at least
 B ☐ (A) three selection signals are required. (B) four selection signals are required. (C) five selection sig-
 C ☐ nals are required. (D) two selection signals are required.
 D ☒

4. A ☐ To store data in a magnetic core, the two binary states (0 and 1) are represented by the
 B ☐ (A) voltage across the output windings. (B) eddy currents in the core. (C) position of the core. (D)
 C ☐ remanent flux densities of the core.
 D ☒

5. A ☐ If the words in a digital system are made up of 35 bits, then a magnetic core storage unit built to store
 B ☒ these words must have
 C ☐ (A) 70 planes. (B) 35 planes. (C) 50 planes. (D) 40 planes.
 D ☐

6. A ☒ When a magnetic core is used as a controllable transformer, at least
 B ☐ (A) 3 windings are required. (B) 4 windings are required. (C) 2 windings are required. (D) 6 windings
 C ☐ are required.
 D ☐

7. A ☐ The speed of domain rotation in a ferromagnetic material is
 B ☒ (A) affected by eddy currents. (B) limited only by the speed with which the small magnetic fields of
 C ☐ single electrons can move. (C) limited by the speed with which the domain walls can move. (D) slower
 D ☐ than that of domain wall motion.

8. A ☐ A Crowe cell provides magnetic shielding between the read and write conductors when
 B ☐ (A) its temperature is above its critical temperature. (B) it is in its normal resistive state. (C) the mag-
 C ☐ netizing force is greater than the critical field strength. (D) it is in its superconductive state.
 D ☒

9. A ☐ In the magnetic drum memory unit of Figure 25, a word is shifted out of BR and recorded on the drum
 B ☒ IMMEDIATELY after
 C ☐ (A) a control pulse is applied to FF₁. (B) EC produces a start pulse. (C) C produces a blocking pulse.
 D ☐ (D) the control unit applies write signals to AND gates 3, 5 and 6.

10. A ☒ Integrated circuit memories
 B ☐ (A) can be obtained with random access read-write and read-only capabilities. (B) are difficult to fabri-
 C ☐ cate in the random access form. (C) will probably never be a very popular memory form because of
 D ☐ inherent limitations. (D) are the only memory units that are compatible with other IC logic elements.

UNIT EXAMINATION FOR COURSE 52

This examination contains a number of different type questions: multiple choice, true or false, fill in the blank and direct question. To assist us in grading, please print your answers clearly on the answer sheet. **COMPLETE THE PORTION STATING YOUR NAME, ADDRESS AND STUDENT NUMBER ON THE ANSWER SHEET AND SEND ONLY THE ANSWER SHEET FOR GRADING.** Retain the examination questions for your records.



BELL & HOWELL SCHOOLS

© COPYRIGHT 1973
BELL & HOWELL SCHOOLS, INC.
ALL RIGHTS RESERVED



1. Electronic switching or gating circuits operate by alternating between two characteristic states or conditions.
2. The use of an unsaturated condition as one of the states in an electronic switching circuit provides for
 (A) maximum possible output voltage variations. (B) highest possible switching speeds. (C) maximum power transfer between stages. (D) highest possible input pulse amplitudes.
3. In switching circuits, the logical one and logical zero are digital representations of the circuit operating in a binary mode.
4. The Boolean function $A + B = Z$ represents the operation of a two-input gate producing a logical one output
 (A) when either or both inputs are logical one. (B) only when both inputs are logical one. (C) only when input A is a logical one. (D) only when input B is a logical one.
5. The accompanying truth table is associated with the OR function.

A	B	C	Z
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Table 1

6. The function $\overline{A} \cdot B = Z$ represents a gate where a logical one at input A inhibits a logical one at the output, regardless of the value at input B.

7. A diode circuit utilizing reverse switching operates between a point in the cutoff region and a point in the breakdown region of the diode characteristic.
8. The primary advantage of a transistor switching circuit is the fact that the transistor can provide signal amplified.
9. A transistor switching circuit arrangement providing the basic OR function provides the IOR function if the transistors are connected in the common-emitter configuration.
10. The largest contribution to transistor turn-on time is
(A) the saturation storage phenomenon. (B) the presence of circuit capacitances.
11. The logical function, implemented by the accompanying diode circuit, is OR.

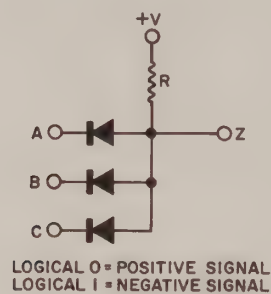


Figure 1

12. The transistor in the accompanying logic circuit
(A) provides the NOT function for the output of the diode arrangement.
(B) establishes the Boolean function for the overall circuit arrangement.
(C) provides diode output amplification without signal inversion.

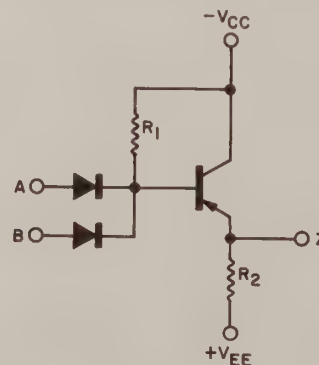


Figure 2

13. With the inputs shown for the accompanying circuit, select the proper output signal.

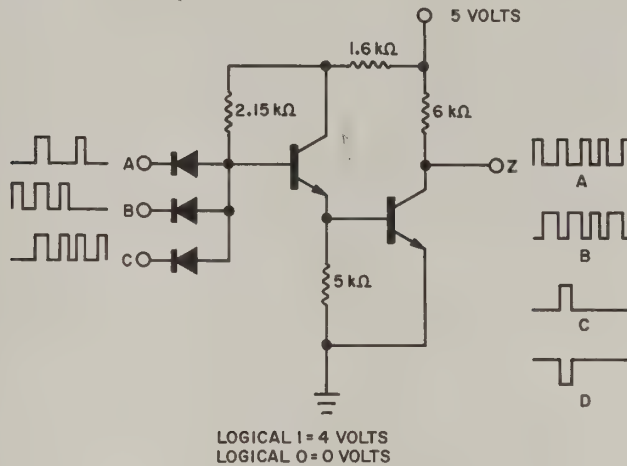


Figure 3

14. Direct-coupled transistor logic is seldom used because of its many disadvantages. True or False?
15. A four-input resistor transistor logic circuit designed to produce the NOR or NAND function will utilize
(A) two transistors. (B) one transistor. (C) four transistors.
16. The capacitor in resistor-capacitor transistor logic
(A) has no effect on circuit logic function. (B) increases circuit switching time. (C) inverts the input signal conditions.
17. The accompanying current-mode logic circuit usually is driven by a
(an) VCM transistor circuit.

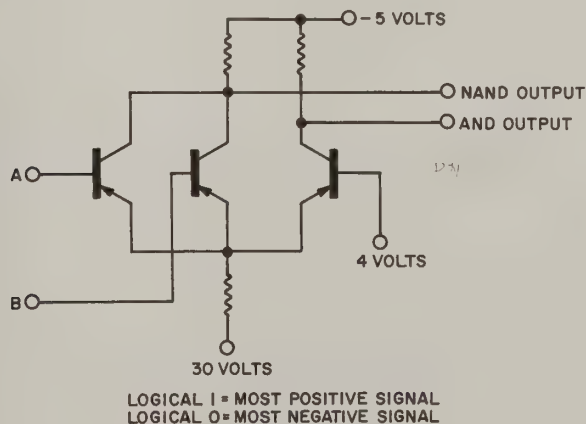


Figure 4

18. When used to specify the logical operation of a switching circuit, the input and output waveforms are known as a timing diagram.

19. The double negation of the OR function results in the OR function.

20. Reversing the logic assignments for the signals of a particular switching circuit results in
(A) a complementary operation. (B) a dual operation.

21. The accompanying circuit produces the exclusive OR function.

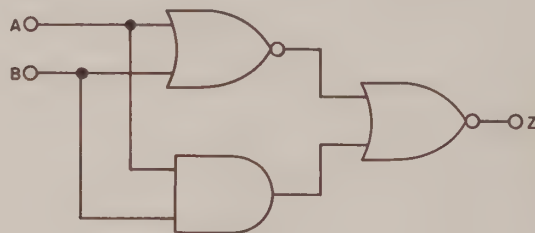


Figure 5

22. Select the appropriate truth table for the following logical relationships: When A and B are both 0, Z must be 1; when A or B is 1, Z must be 1; and when A and B are both 1, Z must be 0.

A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1

A	B	Z
0	0	1
0	1	0
1	0	1
1	1	0

A	B	Z
0	0	1
0	1	1
1	0	0
1	1	0

A	B	Z
0	0	1
0	1	1
1	0	1
1	1	0

(A)

(B)

(C)

(D)

Figure 6

23. If a logic circuit has four input signals, the truth table must contain 16 logical combinations.

24. Write the product-of-sums switching function for the circuit having the accompanying truth table.

A	B	C	D	Z
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	1	0	0	0
0	0	1	1	1
0	1	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	1
1	0	1	1	1
1	1	1	1	1
1	1	0	1	1
1	1	1	0	1

Table 2

25. Simplify the Boolean expression:

$$(A \cdot B \cdot C \cdot D) + (A \cdot B \cdot C \cdot \bar{D}) + (\bar{A} \cdot \bar{B} \cdot C \cdot D)$$

26. The equivalent binary number for decimal 18.1875 is 10010.0011.

27. The binary number 01000 is a four-position BCD code with odd parity.

28. The accompanying encoder circuitry converts decimal numbers to binary numbers using the XS-3 code.

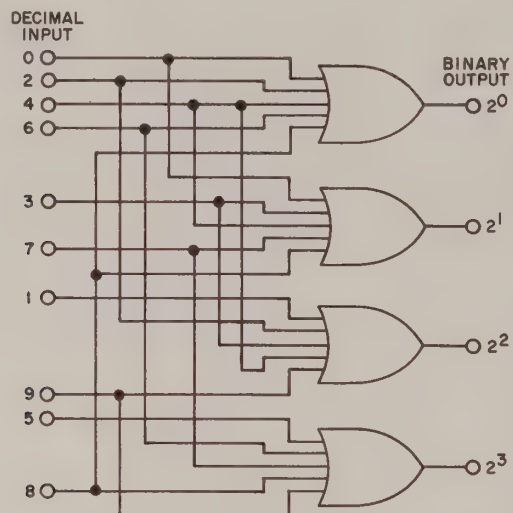


Figure 7

29. The accompanying decoder circuitry converts binary numbers into decimal numbers using the Gray code.

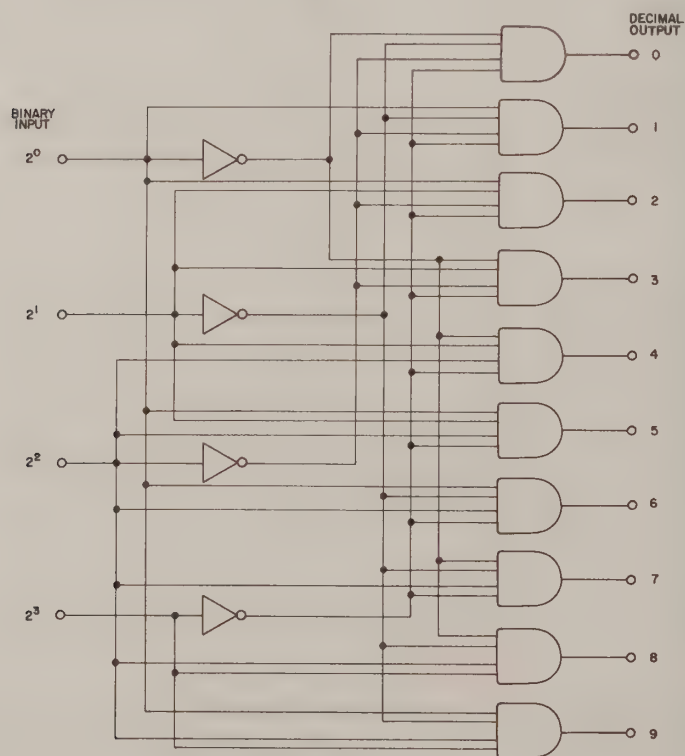


Figure 8

30. The sum output for the accompanying half adder (with the inputs shown) is 011.

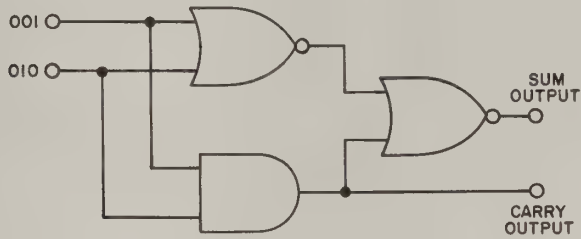


Figure 9

31. Select the diagram illustrating the proper digital conditions at all points in the accompanying full adder.

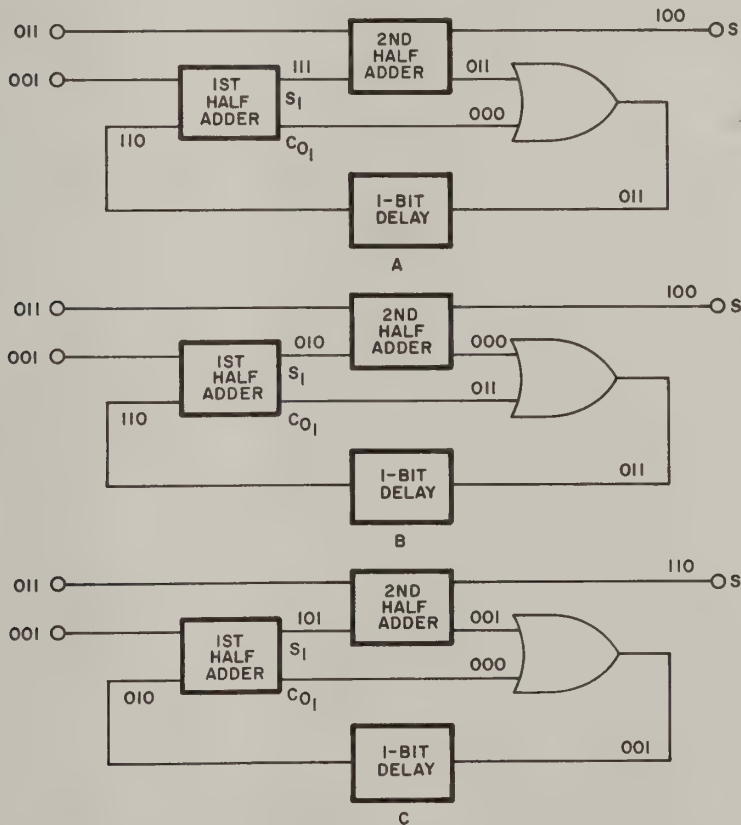


Figure 10

32. Select the diagram illustrating the proper digital conditions at all points in the accompanying full subtractor.

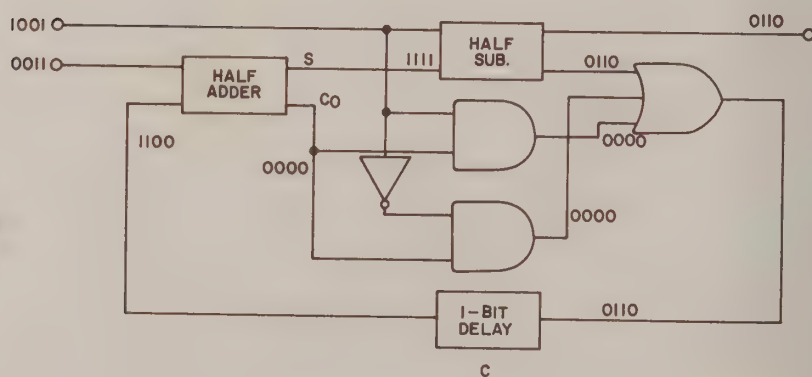
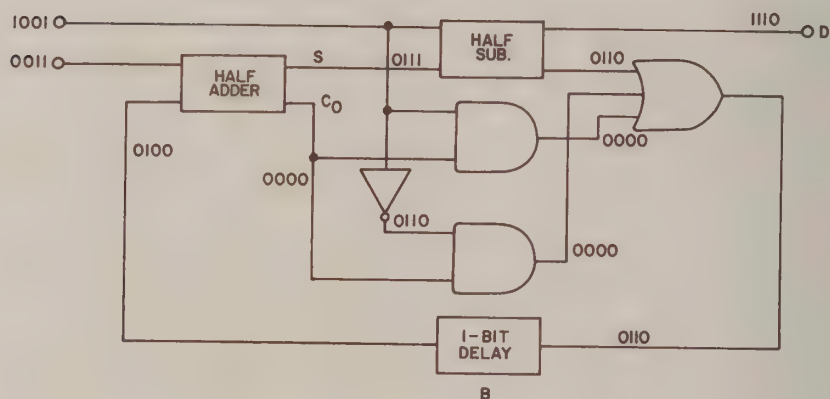
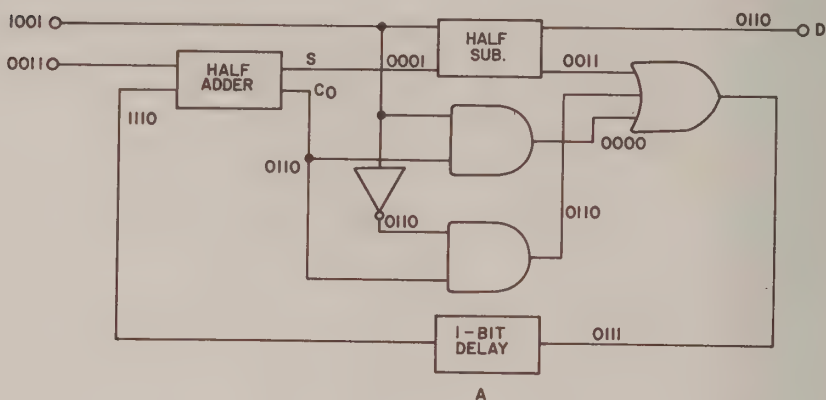
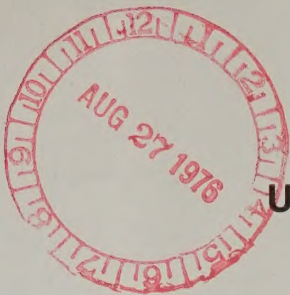


Figure 11

33. The TTL logic system is commonly used in IC logic networks. It provides high noise immunity, a large number of inputs, and high switching speeds.
34. In a particular logic circuit the logical one signal normally is 5 volts. If the voltage falls below 2.8 volts, the circuit switches to the logical zero state. Therefore, the logical one noise margin for the circuit is 2.2 volts.
35. The J-K flip-flop is more practical when constructed using integrated circuits.
36. In a digital system, clock pulses are used to synchronize the operation of the digital components.
37. The smallest register in a digital system using a word length of sixteen bits must have a minimum capacity of 16 bits.
38. In terms of performing mathematical operations with a shift register, shifting three bits to the left corresponds to
 (A) multiplying the register contents by a factor of 8. (B) dividing the register contents by a factor of 3. (C) dividing the register contents by a factor of 8. (D) multiplying the register contents by a factor of 6.
39. The fastest form of data retrieval is obtained by a computer whose memory is based on the IC design.
40. As the size of ferrite cores decreases, the core switching speed decreases. True or False? False



UNIT EXAMINATION 52 ANSWER SHEET

SEP
276

GRADE 88
W. ROZKO
INSTRUCTOR

Name Kent M Bestwick
Address PO Box 329
City FT HUACHUCA
State/Province ARIZONA Zip Code 85613

Student No. K36739 IGW

UE52B AUG 27 1976

Write your answers in the space provided. Be sure to include units, such as volts, μF , etc. where necessary.
RETURN ONLY THE ANSWER SHEET FOR GRADING. Keep the exam for your records.

- | | |
|-------------------------|-------------------------------|
| 1. <u>two</u> | 11. <u>OR</u> |
| 2. <u>B</u> | 12. <u>C</u> |
| 3. <u>state</u> | 13. <u>C</u> |
| 4. <u>A</u> | 14. <u>False</u> |
| 5. <u>NAND</u> | 15. <u>B</u> |
| 6. <u>INHIBITS</u> | 16. <u>A</u> |
| 7. <u>BREAKDOWN</u> | 17. <u>NPN</u> |
| 8. <u>AMPLIFICATION</u> | 18. <u>SWITCHING FUNCTION</u> |
| 9. <u>NOR</u> | 19. <u>OR</u> |
| 10. <u>B</u> | 20. <u>B</u> |

(Continued on reverse side)

88

21. EXCLUSIVE - OR

31. A

22. D

32. C

23. 16

33. TTL

24. $(A+B+C+D)(A+\bar{B}+C+D)(\bar{A}+B+C+D)$

34. 2.2

25. $ABC + \bar{A}\bar{B}CD$

35. J-K

26. 10010.0011

36. SYNCHRONIZE

27. ODD

37. 16

28. XS-3

38. A

29. GRAY

39. IC

30. 011

40. False

